MS™-DOS VERSION 2
REFERENCE GUIDE

COMPAQ®

MS™-DOS VERSION 2
REFERENCE GUIDE

CONTENTS

CHAPTER 1
INTRODUCTION

1-1. MS-DOS: AN OVERVIEW 1-1
1-2. KEY FUNCTIONS IN MS-DOS 1-2
1-3. COPYING DISKETTES:
THE DISKCOPY COMMAND 1-5
Single-Drive Procedure 1-5
Dual-Drive Procedure 1-7
1-4. COMPARING DISKETTES:
THE DISKCOMP COMMAND 1-8
Single-Drive Procedure 1-8
Dual-Drive Procedure 1-10
1-5. ALL ABOUT FILES 1-11
Files on the MS-DOS Diskette 1-11
Syntax Notation 1-13
Naming Files 1-14
Valid Characters 1-15
Valid File Specifications 1-15
Illegal File Specifications 1-16
Reserved Device Names 1-17
Wild Card Characters 1-17
? Character 1-18
* Character 1-18
1-6. PROTECTING YOUR FILES 1-21
1-7. LOADING MS-DOS 1-22
1-8. ENTERING THE DATE AND TIME 1-23
The DATE Prompt 1-23
The TIME Prompt 1-24
1-9. AUTOMATIC PROGRAM EXECUTION 1-26

CHAPTER 2
FDISK UTILITY

2-1. INTRODUCTION 2-1
2-2. INSTALLING AN MS-DOS PARTITION 2-2
As the Entire Fixed Disk 2-3

As One of Several Partitions 2-5
As the First Operating System on the Disk 2-5
With Other Operating Systems Already on the Disk 2-9
2-3. FORMATTING A PARTITION 2-12
2-4. MODIFYING THE ACTIVE PARTITION 2-15
2-5. REMOVING AN MS-DOS PARTITION 2-17
2-6. LISTING PARTITION DATA 2-20
2-7. SELECTING ANOTHER FIXED DISK 2-21

CHAPTER 3
TREE-STRUCTURED DIRECTORIES

3-1. THE DIRECTORY STRUCTURE 3-1
3-2. PATHING 3-3
Pathnames 3-3
Working Directory Notations 3-4
3-3. DIRECTORY COMMANDS 3-6
Creating a Subdirectory: The MKDIR Command 3-6
Listing Files in a Working Directory: The DIR Command 3-7
Changing a Working Directory: The CHDIR Command 3-8
Deleting a Directory: The RMDIR Command 3-9
Displaying the Directory Structure: The TREE Command 3-9

CHAPTER 4
MS-DOS COMMANDS

4-1. INTRODUCTION 4-1
4-2. TYPES OF MS-DOS COMMANDS 4-2
4-3. COMMAND OPTIONS 4-4
4-4. RULES FOR USING MS-DOS COMMANDS 4-5
4-5. BATCH PROCESSING 4-7
The AUTOEXE.BAT File 4-7
Creating a Batch File With Replaceable Parameters . . . 4-9
Executing a Batch File 4-10
Batch Commands 4-11

4-6. INPUT AND OUTPUT 4-13
Redirecting Output 4-13
Redirecting Input 4-14
Filters 4-14
Command Piping 4-15
4-7. HOW TO CONFIGURE YOUR SYSTEM 4-16
Configuration Commands 4-16
4-8. COMMANDS 4-18
ANSI.SYS Device Driver 4-19
  Positioning the Cursor 4-20
  Screen Display Options 4-22
ASSIGN Command 4-26
BACKUP Command 4-27
BREAK Command 4-30
BUFFERS Command 4-31
CHDIR (CHANGE DIRECTORY) Command 4-32
CHKDSK (CHECK DISK) Command 4-33
CLS Command 4-36
COMP (COMPARE) Command 4-37
COPY Command 4-39
  Wild Card Characters 4-40
  Special Considerations of Concatenation 4-45
  Changing the Date of a File 4-46
CTTY Command 4-47
DATE Command 4-48
DEL (DELETE) Command 4-49
DEVICE Command 4-50
DIR (DIRECTORY) Command 4-51
  Wild Card Characters 4-52
  Listing All Files 4-52
  Listing Selected Files 4-53
DISKCOMP (DISKETTE COMPARE) Command 4-54
DISKCOPY (DISKETTE COPY) Command 4-57
  Diskette Fragmentation Problems 4-58
ECHO Command 4-60
ERASE Command 4-61
EXE2BIN Command 4-62
FDISK Utility 4-64

FILES Command 4-65
FIND Command 4-66
FOR Command 4-68
FORMAT Command 4-70
GOTO Command 4-74
GRAPHICS Command 4-75
IF Command 4-76
MKDIR (MAKE DIRECTORY) Command 4-77
MODE Command 4-78
MORE Command 4-85
PATH Command 4-86
PAUSE Command 4-88
PRINT Command 4-90
PROMPT Command 4-93
RECOVER Command 4-95
REM (REMARK) Command 4-97
REN (RENAME) Command 4-98
RESTORE Command 4-99
RMDIR (REMOVE DIRECTORY) Command 4-103
SET Command 4-104
SHELL Command 4-106
SHIFT Command 4-107
SORT Command 4-108
SYS (SYSTEM) Command 4-110
TIME Command 4-111
TREE Command 4-113
TYPE Command 4-114
VER (VERSION) Command 4-115
VERIFY Command 4-116
VOL (VOLUME) Command 4-117

CHAPTER 5
EDLIN

5-1. INTRODUCTION 5-1
5-2. RULES FOR USING EDLIN 5-3
5-3. EDLIN COMMANDS 5-5
Intraline Editing 5-6

    Standard Editing Keys 5-6
    CTRL Key and Multiple-Key Functions 5-7
    Special Function Keys 5-8
  Interline Editing Commands 5-9
    Command Parameters 5-10
    APPEND (A) Command 5-13
    COPY (C) Command 5-14
    DELETE (D) Command 5-17
    EDIT Command 5-20
    END (E) Command 5-22
    INSERT (I) Command 5-24
    LIST (L) Command 5-29
    MOVE (M) Command 5-32
    PAGE (P) Command 5-34
    QUIT (Q) Command 5-37
    REPLACE (R) Command 5-38
    SEARCH (S) Command 5-43
    TRANSFER (T) Command 5-46
    WRITE (W) Command 5-48
  Special Function Keys 5-49
    F1 Key (Copy 1) 5-50
    F2 Key (Copy To) 5-51
    F3 Key (Copy All) 5-52
    DEL (DELETE) Key (Skip 1) 5-53
    F4 Key (Skip To) 5-54
    ESC (ESCAPE) Key (Skip All) 5-55
    INS (INSERT) Key (Add In) 5-56
    F5 Key (New Temp) 5-57
    F6 Key (^Z) 5-58

CHAPTER 6
MS-LINK

6-1. OVERVIEW 6-1
  Capabilities of the Linker Utility 6-1
  MS-LINK Operation Summary 6-1
  Definitions 6-3
6-2. COMBINING AND ARRANGING SEGMENTS 6-5
6-3. FILES USED BY MS-LINK 6-9
  MS-LINK Input Files 6-9

MS-LINK Output Files 6-9
The VM.TMP File. 6-10
6-4. RUNNING MS-LINK 6-11
Command Prompts. 6-11
Switches 6-13
/DSALLOCATE 6-14
/HIGH 6-14
/LINENUMBERS 6-15
/MAP 6-15
/PAUSE 6-15
/STACK:*number*. 6-16
Command Characters. 6-16
Invoking MS-LINK 6-18

CHAPTER 7
DEBUG

7-1. INTRODUCTION. 7-1
7-2. STARTING DEBUG 7-2
7-3. DEBUG COMMANDS 7-3
DEBUG Commands Summary. 7-4
Parameters. 7-5
A (ASSEMBLE) Command 7-9
C (COMPARE) Command 7-12
D (DUMP) Command 7-13
E (ENTER) Command 7-16
F (FILL) Command 7-18
G (GO) Command. 7-19
H (HEX) Command 7-21
I (INPUT) Command 7-22
L (LOAD) Command 7-23
M (MOVE) Command 7-25
N (NAME) Command 7-26
O (OUTPUT) Command 7-29
Q (QUIT) Command 7-30
R (REGISTER) Command 7-31
S (SEARCH) Command 7-34
T (TRACE) Command 7-35
U (UNASSEMBLE) Command 7-36
W (WRITE) Command. 7-39

APPENDIX A
MESSAGES

APPENDIX B
INDEX

ILLUSTRATIONS

FIGURES:

3-1. A Tree-Structured Directory 3-1
3-2. Directories on the Disk in Drive A: 3-12
6-1. MS-LINK Operation Block Diagram 6-2
6-2. Offset Addressing of Items From a Single Base Address 6-7

TABLES:

6-1. Summary of Methods Used to Invoke MS-LINK. 6-18
6-2. Command Prompt Summary 6-19
6-3. MS-LINK Switch Parameters 6-20
7-1. (Partial) Logical to Physical Mapping Table 7-6
7-2. Set and Clear Codes for DEBUG Flags 7-32

SCOPE AND PURPOSE OF THIS MANUAL

This book is designed as a reference tool for using the MS™-DOS operating system. It describes the basic functions of MS-DOS, such as creating, editing, executing, linking, and debugging programs.

It is designed to complement the information presented in your COMPAQ Computer ***OPERATIONS GUIDE***. For more information on the operation of your computer, see your ***OPERATIONS GUIDE***.

ORGANIZATION OF THE MS-DOS REFERENCE GUIDE

The ***MS-DOS VERSION 2 REFERENCE GUIDE*** is organized into seven chapters and three appendices.

Chapter 1, INTRODUCTION, describes the MS-DOS operating system and its importance. It also gives basic information about MS-DOS functions and provides background information on loading the operating system.

Chapter 2, FDISK UTILITY, describes the FDISK utility and the methods used to install, modify, and remove an MS-DOS partition, and to list partition information.

Chapter 3, TREE-STRUCTURED DIRECTORIES, describes hierarchical directories and the commands used to list, create, locate, change, and delete directories.

Chapter 4, MS-DOS COMMANDS, describes the MS-DOS commands. Information on options, batch processing, and system configuration is included.

Chapter 5, EDLIN, describes the text line editor used for editing files, and the rules and commands for its use.

Chapter 6, MSTM-LINK, describes the utility used for linking programs to allow you to increase the potential of your applications.

Chapter 7, DEBUG, describes the utility used to test and modify programs that are being developed.

Appendix A, MESSAGES, lists the messages that MS-DOS produces, with brief descriptions and their sources.

Appendix B, INDEX, supplies page locations for specific information in this reference manual.

CHAPTER 1

INTRODUCTION

1-1. MS-DOS: AN OVERVIEW . . . 1-1
1-2. KEY FUNCTIONS IN MS-DOS . . . 1-2
1-3. COPYING DISKETTES: THE DISKCOPY COMMAND . 1-5
  Single-Drive Procedure . . . 1-5
  Dual-Drive Procedure . . . 1-7
1-4. COMPARING DISKETTES: THE DISKCOMP COMMAND . . . 1-8
  Single-Drive Procedure . . . 1-8
  Dual-Drive Procedure . . . 1-10
1-5. ALL ABOUT FILES . . . 1-11
  Files on the MS-DOS Diskette . . . 1-11
  Syntax Notation . . . 1-13
  Naming Files . . . 1-14
    Valid Characters . . . 1-15
    Valid File Specifications . . . 1-15
    Illegal File Specifications . . . 1-16
    Reserved Device Names . . . 1-17
    Wild Card Characters . . . 1-17
      ? Character . . . 1-18
      * Character . . . 1-18
1-6. PROTECTING YOUR FILES . . . 1-21
1-7. LOADING MS-DOS . . . 1-22
1-8. ENTERING THE DATE AND TIME . . . 1-23
  The DATE Prompt . . . 1-23
  The TIME Prompt . . . 1-24
1-9. AUTOMATIC PROGRAM EXECUTION . . . 1-26

1-1. MS-DOS: AN OVERVIEW

MS-DOS, an acronym for Microsoft Disk Operating System, is a set of software programs designed for 8086/8088-based computers.

An operating system is what operates the computer. It interfaces with you, applications software, and the hardware to allow you to enter instructions through the keyboard. These instructions interact with the memory, disk drives, and input/output devices to allow you to create, store, and recall files, run and link programs, and access printers and disk drives.

Unlike earlier versions of Microsoft's DOS, MS-DOS VERSION 2 supports the use of a fixed disk. Fixed disks have a much greater storage capacity, permit several users to work with the system, store data in separate directories, and allow more than one operating system to be installed in a system.

MS-DOS VERSION 2 also formats diskettes with nine sectors per track, rather than eight, which creates formatted diskettes with a capacity of 360 Kbytes, rather than 320 Kbytes.

In this book, the term "diskette" always refers to a 5¼-inch floppy diskette, and the term "fixed disk" always refers to an installed hard disk. The term "disk" is used in this manual to indicate either a diskette or a fixed disk.

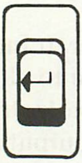
MS-DOS assigns drive designators to the disk drives in your system. It always assigns Drives A: and B: as floppy disk drives, even if your system has one floppy disk drive. If your system has a fixed disk drive, it is assigned Drive C:.

When used to describe diskette operations, the term "single-drive" refers to a system with only one floppy disk drive, even if the system also contains a fixed disk drive. The term "dual-drive" refers to a system with two floppy disk drives.

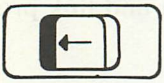
1-2. KEY FUNCTIONS IN MS-DOS

The following keys are frequently used to perform functions when using MS-DOS.

Key: **Description:**



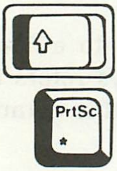
The ENTER key enters a command that has been typed.



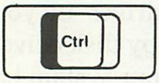
The BACKSPACE key moves the cursor to the left, deleting one character per keystroke.

The CAPS LOCK key, which is a toggle key, locks the keyboard into upper-case letters. To return to the lower case, press the key a second time.

The ESCAPE (ESC) key cancels the line you are on, and moves the cursor down one line. It leaves a backslash (/) on the cancelled line.



When you press the SHIFT key and the PRINTSCREEN (PRTSC) key simultaneously (SHIFT + PRTSC), the printer prints everything displayed on the screen.



The CONTROL (CTRL) key is always used in conjunction with one or two other keys to perform a particular function in an applications program. ^PRTSC (CTRL + PRTSC), for example, causes whatever you type to be echoed to the printer. Pressing ^PRTSC again turns the echoing off.

In this manual, a caret (^) preceding the name of a key means that you press and hold the CTRL key, and then press the key indicated.

For example, if the letter T is to be used as a control character, it appears in text as ^T.

Some other CTRL key functions you might use are:

^ALT + DEL	System reset.
^BREAK	Terminates the operation in progress.
^ENTER	Moves the cursor to the next line so that you can continue typing.
^NUMLOCK	Pauses the current operation; for example, if a large amount of data is being output to the screen, you can stop the scrolling by pressing ^NUMLOCK. Pressing any key starts the scrolling again.

The F1 key copies and displays the previously-entered line, one character at a time.

The F2 key, followed by a character from a previously-entered line, copies and displays the previously-entered line up to the first occurrence of the specified character.

The F3 key copies and displays the previously-entered line in its entirety.

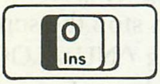
The F4 key, followed by a character from the previously-entered line, copies (but does not display) the first occurrence of the specified character, plus the remainder of the line. Press F3 to display what the F4 key copied.

The F5 key moves the cursor down one line to allow you to edit or retype the currently-displayed line.

The F6 key sends a ^Z character to MS-DOS. A ^Z signals the end of input. When you have finished inputting, press F6 and then ENTER.

The F7 key sends a ^@ (NUL) character to MS-DOS. NUL is used as a string terminator or as a record filler.

The DEL key deletes, or skips over, characters in the previously-entered line. Press F3 to display the remainder of the line. You can press DEL several times to delete as many characters as you want from the previously-entered line.



The INS key turns on the insert mode. This allows you to copy part of the previously-entered line, and insert additional characters where you want them. Pressing INS again cancels the insert mode.

NOTE: When you are instructed to enter data, type the indicated text and then press the ENTER key.

1-3. COPYING DISKETTES: THE DISKCOPY COMMAND

The MS-DOS diskette is essential to operation. It, therefore, has a protective adhesive tab over the write-protect notch. Information cannot be entered on it, and it cannot be used for storage. For these reasons, make one or more back-up copies and store the original safely where it will not be exposed to static electricity, magnetic fields, extreme temperatures or humidity, or subjected to excessive handling or abuse.

To copy your MS-DOS diskette, use the DISKCOPY command and follow the procedure below. DISKCOPY is used to copy the entire contents of one diskette onto another diskette.

NOTE: The following procedure is for a single-drive system. If you have a dual-drive system, your procedure immediately follows this one.

Single-Drive Procedure

Make sure the A> appears on your screen. Enter:

```
DISKCOPY
```

The screen displays:

```
Insert source diskette in drive A:

Strike any key when ready
```

Insert the source diskette into the drive. In this case, your MS-DOS diskette is the source diskette. Once the MS-DOS (source) diskette is inserted, press any key to begin. The system reads as much information as it can into memory and then prompts:

```
Insert target diskette in drive A:
Strike any key when ready

Copying 9 sectors per track, 2 side(s)
```

Replace the source (MS-DOS) diskette that is in the drive with the target diskette. In this case, a blank diskette is your target diskette. Once the blank (target) diskette is inserted, press any key. The data that was pulled from the source diskette into memory is now transferred from memory onto the target diskette. The system then again prompts you to insert the source diskette, in order to pull more data into memory. The process of copying from and onto alternating diskettes continues until the copy is complete. Then the screen displays:

```
Copy complete

Copy another    (Y/N)?
```

Press N to terminate the copy session.

Dual-Drive Procedure

Make sure the A> appears on your screen. Enter:

```
DISKCOPY A: B:
```

The screen displays:

```
Insert source diskette in drive A:

Insert target diskette in drive B:

Strike any key when ready


Copying 9 sectors per track, 2 side(s)
```

Insert the diskettes, if they are not already in place. In this case, your MS-DOS diskette is the source diskette and a blank diskette is your target diskette. Once the diskettes are inserted, press any key to begin. Blocks of data are transferred from the source diskette into memory, and then to the target diskette. This process continues until the copy procedure is finished. Then the screen displays:

```
Copy complete

Copy another   (Y/N)?
```

Press N to terminate the copy session.

1-4. COMPARING DISKETTES: THE DISKCOMP COMMAND

After the DISKCOPY procedure, compare the diskettes to make sure they are identical.

NOTE: The following procedure is for a single-drive system. If you have a dual-drive system, your procedure immediately follows this one.

Single-Drive Procedure

Make sure the A> appears on your screen. Enter:

```
DISKCOMP
```

The screen displays:

```
Insert first diskette in drive A:
Strike any key when ready
```

Insert the first diskette into the drive. In this case, your MS-DOS diskette is the first diskette. Once the MS-DOS (first) diskette is inserted, press any key to begin. The system reads as much information as it can into memory and then prompts:

```
Insert second diskette in drive A:

Strike any key when ready


Comparing 9 sectors per track, 2 side(s)
```

Replace the first (MS-DOS) diskette that is in the drive with the second (copied) diskette. In this case, the copied diskette is your second diskette. Once the copied (second) diskette is inserted, press any key. The data that was pulled from the first diskette into memory is now compared to the data on the second diskette. The system then again prompts you to insert the first diskette, in order to pull more data into memory. The process of reading alternating diskettes continues until the comparison is complete. Then the screen displays:

```
Diskettes compare ok

Compare more diskettes    (Y/N)?
```

Press N to terminate the compare session.

Dual-Drive Procedure

Make sure the A> appears on your screen. Enter:

Press N to terminate the compare session.

```
DISKCOMP A: B:
```

The screen displays:

```
Insert first diskette in drive A:

Insert second diskette in drive B:

Strike any key when ready


Comparing 9 sectors per track, 2 side(s)
```

When the comparison is complete, the screen displays:

```
Diskettes compare ok

Compare more diskettes    (Y/N)?
```

Press N to terminate the compare session.

If the diskettes fail to compare, error messages appear that tell you which areas don't compare.

1-5. ALL ABOUT FILES

FILES ON THE MS-DOS DISKETTE

The MS-DOS program diskette for the COMPAQ computers includes the files listed below.

File Specification:	**Description or Function:**
ANSI.SYS	Device driver used to implement extended screen and keyboard functions.
ASSIGN.COM	Sends requests to alternate drives.
BACKUP.COM	Copies files from a fixed disk to a diskette.
BASIC.COM BASICA.COM BASICA.EXE	The files that make up the BASIC interpreter.
CHKDSK.COM	Reports disk status.
COMMAND.COM	MS-DOS command processor.
COMP.COM	Compares files.
DEBUG.COM	Debugs programs.
DEMO.BAS DEMOX.BAS DEMO.BAT	Main files that execute the various demonstration programs.
DISKCOMP.COM	Compares entire diskettes.
DISKCOPY.COM	Copies entire diskettes.

File Specification:	Description or Function:
EDLIN.COM	Line editor.
EXE2BIN.EXE	Converts .EXE files to .COM files.
FDISK.COM	Installs an MS-DOS partition on a fixed disk.
FIND.EXE	Finds text strings in specified files or standard output.
FORMAT.COM	Formats disks.
GRAPHICS.COM	Prints graphics from the screen.
INTEREST.BAS	Sample BASIC program.
LINK.EXE	Linker utility.
MODE.COM	Utility used to set screen, printer, communications, and system parameters.
MORE.COM	Displays output one screen at a time.
PRINT.COM	Allows data files to be printed while processing other MS-DOS commands.
RECOVER.COM	Recovers lost data on disks.
RESTORE.COM	Reloads diskette files to a fixed disk.
SORT.EXE	Sorts text within a file.
SYS.COM	Transfers MS-DOS system files.
TEST.EXE	User Diagnostics.

File Specification:	Description or Function:
TREE.COM	Displays paths within directories.
WORDS	Sample text file.

The two system files are hidden files, and do not appear in any directory listing.

SYNTAX NOTATION

All MS-DOS commands and statements use the following syntax:

- Items in brackets ([ ]) are optional. For example, in the statement [*d:*]*filename*[.*ext*], the drive designator (*d:*) and extension (.*ext*) are optional.
- Lower-case text in *italics* indicates a variable. The variables that are not in square brackets must be supplied. For example, for *filespec*, type in the name of the file.
- Upper-case words indicate a key or a command name. For example, CTRL indicates the CONTROL key.
- Items in braces ({ }) are entry choices. Enter one of these, unless the entries are also enclosed in brackets ([ ]).
- Items followed by ellipses (...) may be repeated as often as necessary.
- An item followed by a broken bar ¦ is an OR statement in a command. When used with MS-DOS filters, the bar indicates a pipe (see Section 4-6, INPUT AND OUTPUT).
- Punctuation, such as commas, colons, slashes, and equal signs must be included where shown, unless indicated as an option.

NAMING FILES

As many as 112 directory entries may be recorded on a double-sided diskette. All files within the same directory must have unique names so that MS-DOS can distinguish between them.

The name of a file can consist of as many as four parts: a drive designator, which specifies the disk drive; a pathname, which specifies the directory (see Chapter 3, TREE-STRUCTURED DIRECTORIES); a filename, which indicates the name of the file; and an extension, which allows you to categorize files into smaller, more manageable parts.

In MS-DOS, the drive designator is always a letter followed by a colon, such as A: or B:. The filename consists of a maximum of eight characters, and the optional extension consists of a period and three characters. No blank characters can appear between the drive designator and the filename, or the filename and the extension. When a drive designator and extension are used, the colon and period must appear in the appropriate places. For example, A:NEWFILE.EXE is a file located on Drive A:, named NEWFILE, with an extension of .EXE.

The term "file specification" is used to designate a filename and its extension, if one is included. Its shortened version, *filespec,* is frequently used in syntax statements.

When referring to a file specification that contains an extension, always include the extension and the filename in the file specification. If you don't, MS-DOS will be unable to locate the file.

Valid Characters

Characters that are valid for filenames are also valid for extensions. Any other characters are invalid. The valid characters that can be used in filenames and extensions are:

A-Z (upper or lower case)
0-9
$ ( ) & ’ — @
# { } % ~ ! —

No other characters can be used.

NOTE: With MS-DOS VERSION 2 (unlike earlier DOS versions), the characters >, <, \, and ¦ cannot be used in file specifications.

Valid File Specifications

File specifications are usually selected to reflect the file’s contents. Some examples of valid file specifications, and brief explanations of what their contents might be, are listed following:

File Specification:	Contents of the File:
NTH-NAME	A program designed to extract every 2nd, 3rd, 4th, or Nth name from a large master mailing list, and use those names to create a smaller list with a similar demographic character.
FRIENDS	A list of names, addresses, and birthdays of your friends.
ACCTS.REC	A list of current accounts receivable, including names, addresses, amounts due, and status. You might use a file named ACCTS.PAY in conjunction with this file to list current accounts payable.

File Specification:	Contents of the File:
GET-30S	A program that extracts those accounts receivable which have been outstanding for 30 days.
SUBS	A list of subscribers to a club publication.
FUNSTERS	An educational word game that you have made up for your children.
QUANBRK	A program that calculates price discounts associated with the purchase of a particular quantity of goods.
ITIN.MAR	An itinerary for a trip you plan to take in March.

Illegal File Specifications

Some file specifications are unacceptable because of spacing, length, or improperly used punctuation marks and symbols. Some examples of file specifications and the reasons for their unacceptability appear following:

Unacceptable File Specification:	Reason:
BOOK I	Space not allowed.
FERGESON'S.LST	Filename too long.
STURGIS,M	Comma not allowed.
STURGIS:M	Colon improperly used.
.STU	Missing filename.
DAN'S.EPIC	Extension too long.

Reserved Device Names

MS-DOS reserves the names listed below for some specific input/output devices. Do not use these three-letter combinations as filenames or extensions.

In these names, drive designators and extensions are ignored, and colons are optional.

File Specification:	**Description:**
AUX: or COM1:	Refers to the primary asynchronous communications adapter port.
COM2:	Refers to secondary asynchronous communications adapter port.
CON:	Refers to keyboard input and screen output.
LPT1: or PRN:	Refers to the primary line printer (output device only).
LPT2: and LPT3:	Refers to another line printer (output device only).

NOTE: Reserved names may be used in place of file specifications in appropriate commands and string statements.

Wild Card Characters

The wild card characters, ? (question mark) and * (asterisk), are a short notation for many multiple file applications. Using these characters provides greater flexibility in MS-DOS commands.

? Character

A ? character in a filename or extension indicates that any single, valid character may occupy that position. For example:

```
DIR BUD?DEPT.83
```

lists all directory entries (in the default drive) beginning with BUD, having any next character, and followed by DEPT, with an extension of .83. Files that might be listed by this DIR command are:

```
BUD1DEPT.83
BUD3DEPT.83
BUD9DEPT.83
```

This listing could indicate that departments 1, 3, and 9 have their 1983 budgets listed on this diskette.

As with other acceptable characters, a maximum of eight ?s can be used in the filename, and a maximum of three ?s can be used in the extension.

* Character

A * character in a filename or extension indicates that any acceptable number (up to eight in the filename and up to three in the extension) of valid characters may occupy that position and all following positions. For example:

```
DIR D115*.'83
```

lists all directory entries (in the default drive) beginning with D115, having any next characters (up to four), and with an extension of .'83.

The following files might be listed by this DIR command:

```
D115LAB.'83
D115EXP.'83
D115FCST.'83
```

This list could indicate that department 115 has labor (LAB), expense (EXP), and monthly forecast (FCST) reports for 1983 ('83) listed on this diskette.

NOTE: *.* indicates all files on a diskette. Using the command DEL *.* deletes all files in the default directory on the disk in the default drive, regardless of filename or extension. Use *.* very carefully.

The following are example uses of the ? and * characters:

Example 1:

To delete the directory entries for all files named FORECAST on Drive A: (regardless of extension), use the DEL command like this:

```
DEL A:FORECAST.???
or
DEL A:FORECAST.*
```

The list of deleted entries might look like this:

```
FORECAST.115
FORECAST.117
FORECAST.250
FORECAST.300
```

This list might indicate the departments that have submitted their monthly forecasts.

Example 2:

To list the directory entries for all files on Drive B: (regardless of filenames) with an extension of .JUN, enter:

```
DIR B:????????.JUN
or
DIR B:*.JUN
```

The entries listed might look like this:

```
ACCTSREC.JUN
ACCTSPAY.JUN
FORECAST.JUN
EXPENSES.JUN
```

This list could indicate the monetary figures that have been entered for the month of June.

Example 3:

To list the directory entries for all files on Drive B: beginning with 100, and with extensions beginning with 1Q, enter:

```
DIR B: 100?????.1Q?
or
DIR B: 100*.1Q*
```

The entries listed might indicate, for example, that department 100 has the following files:

100EXP.1Q	(expense totals for the first quarter)
100MAN.1Q	(manpower totals for the first quarter)
100MAN.1QJ	(manpower requirements for January)
100MAN.1QM	(manpower requirements for March)
100EXP.1QF	(expense totals for February)

1-6. PROTECTING YOUR FILES

If you are processing information that is irreplaceable or demands a high degree of security, protect it from unauthorized use or accidental destruction by:

- Removing your diskettes from the drives when not in use.
- Copying diskettes to serve as back-up information.
- Using your COMPAQ computer in a secure area.

1-7. LOADING MS-DOS

To load MS-DOS, remove the diskette from the pocket and the jacket in the front of the book. Open the door to the left-hand disk drive; it is known as Drive A:. Holding the diskette with the printed label facing left, insert it into the drive. Close the door and turn the computer on, or do a system reset (^ALT + DEL).

1-8. ENTERING THE DATE AND TIME

When you turn your COMPAQ computer on, MS-DOS searches for the COMMAND.COM file and loads it into memory. Once COMMAND.COM is loaded, the screen displays first a date and then a time prompt. If you want the computer to use the current date and time when constructing directories, supply it with the correct information by using the following procedures.

THE DATE PROMPT

The screen displays:

```
Current date is Tue 1-01-1980

Enter new date:
```

To bypass the date prompt, press ENTER. The computer uses the date displayed as the date default value. To set the current date (for example 12-22-83), type the month, date, and year, and press ENTER. The month and date can be one- or two-digit numbers, and the year can be two- or four-digit numbers. Dashes (–) or slashes (/) can be used to separate the entries. Any valid date will be accepted, as long as you follow the format.

After you have entered or bypassed the date prompt, the screen then displays the time prompt.

THE TIME PROMPT

The screen displays:

```
Current time is 0:00:11.58
Enter new time:
```

The time displayed, in hours, minutes, seconds, and hundredths of seconds, is the current time that has elapsed since you turned the computer on.

To bypass the time, press ENTER. To set the computer's built-in clock, type the hours and minutes (for example, 6:30), and press ENTER. You must enter the time in military hours (for example, 6:30 pm = 18:30 in military hours). It is not necessary to enter seconds and hundredths of seconds. After you have entered or bypassed the time prompt, the screen then displays:

```
The COMPAQ Personal Computer MS-DOS
Version 2.XX

(C) Copyright COMPAQ Computer Corp. 1983

(C) Copyright Microsoft 1981, 82, 83

A>
```

When the A> prompt appears, MS-DOS is loaded into memory and ready to accept commands. If you want the system to default to Drive B: instead of Drive A:, you must change the default drive to Drive B:. To change the default drive to Drive B:, enter:

```
B:
```

The screen displays:

```
A>B:

B>
```

Drive B: is now established as the default drive.

1-9. AUTOMATIC PROGRAM EXECUTION

If you want the computer to execute additional instructions when you start MS-DOS, you can create an AUTOEXEC.BAT file and include these instructions in it. For example, you might want a certain applications program to start up every time you turn your computer on.

When you first turn your computer on, MS-DOS loads COMMAND.COM into memory. During this process, it searches for an AUTOEXEC.BAT file on the disk. If it finds the AUTOEXEC.BAT file, and the file contains specific instructions, such as to load BASIC and run a program called MENU, the computer acts on these instructions. The computer bypasses the date and time prompts described above, unless they are included in the AUTOEXEC.BAT file. If there isn't an AUTOEXEC.BAT file, the date and time prompts are displayed.

For instructions on creating an AUTOEXEC.BAT file, see Chapter 4, MS-DOS COMMANDS.

CHAPTER 2

FDISK UTILITY

2-1. INTRODUCTION 2-1
2-2. INSTALLING AN MS-DOS PARTITION 2-2
  As the Entire Fixed Disk 2-3
  As One of Several Partitions 2-5
    As the First Operating System on the Disk. 2-5
    With Other Operating Systems Already on the Disk . . . 2-9
2-3. FORMATTING A PARTITION. 2-12
2-4. MODIFYING THE ACTIVE PARTITION 2-15
2-5. REMOVING AN MS-DOS PARTITION 2-17
2-6. LISTING PARTITION DATA 2-20
2-7. SELECTING ANOTHER FIXED DISK. 2-21

2-1. INTRODUCTION

Fixed disks are also known as hard or rigid disks. Their principal advantages over floppy diskettes are greater storage capacity and quicker response time.

The COMPAQ Computer fixed disk has a formatted capacity of 10 megabytes, or 10 million characters. That is, it stores about as much data as 27 double-sided, double-density diskettes formatted with nine sectors.

Before a fixed disk can be used, you must use the FDISK and FORMAT utilities to set up the fixed disk so it can be accessed by MS-DOS. The FDISK utility is used to specify drives and install partitions. The FORMAT utility initializes the directory and sets up the file allocation table.

The fixed disk drive is always assigned as Drive C:.

2-2. INSTALLING AN MS-DOS PARTITION

Partitions are areas on a fixed disk that are set aside for a specific purpose, such as a particular operating system. A fixed disk can have one to four partitions.

To install, modify, or remove partitions, to list partition information, or to select another fixed disk drive, you must first access the FDISK menu. To do this, insert the MS-DOS diskette in Drive A: and close the door. Turn your computer on, or do a system reset, so that the A> prompt is displayed. Now enter:

```
FDISK
```

If there is no fixed disk, the screen displays an error message. Otherwise, the screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* FDISK Menu *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Choose one of these options:
   1. Install MS-DOS Partition
   2. Modify Active Partition
   3. Remove MS-DOS Partition
   4. List Partition Information

Enter your choice:    [1]

Press ESC to return to MS-DOS
```

NOTE: A fifth option appears on this menu if your computer is configured with more than one fixed disk. If you want to select a second fixed disk now, proceed to Section 2-7, SELECTING ANOTHER FIXED DISK.

Enter the number corresponding to the task you want to perform. The default choice is 1. Each task is discussed in the following sections.

There are three methods of installing an MS-DOS partition: 1) with the entire fixed disk as one MS-DOS partition; 2) as one of several partitions with no other operating systems installed on the disk; and 3) as one of several partitions with other operating systems already installed in the other partitions.

Each of these methods is detailed in this section.

AS THE ENTIRE FIXED DISK

If you want to install an MS-DOS partition as the entire fixed disk, simply press ENTER (1 is the default value).

If an MS-DOS partition has already been created, the screen displays an error message. Otherwise, the screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Install MS-DOS Partition *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Do you wish to use the entire
fixed disk for MS-DOS (Y/N)?   [Y]

Press ESC to return to FDISK Menu
```

Since you want to use the entire fixed disk for an MS-DOS partition, the answer is yes. Press ENTER since Y is the default response for this prompt.

FDISK then allocates all the disk space to MS-DOS. The screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Install MS-DOS Partition *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

An MS-DOS partition has been installed.
A system reset will occur to update MS-DOS.
The MS-DOS partition must be formatted using
the FORMAT command after the system reset.

Insert MS-DOS diskette in drive A.
Press ENTER when ready:    [   ]

Press ESC to return to FDISK Menu
```

After you press ENTER, a system reset occurs, and the A> prompt appears on the screen. Now you need to format the partition. Proceed to Section 2-3, FORMATTING A PARTITION.

AS ONE OF SEVERAL PARTITIONS

When MS-DOS is installed as one of several partitions, the procedure used depends on whether other operating systems are already installed in other partitions. In both instances, however, the fixed disk must be told the size and location of the MS-DOS partition you are installing.

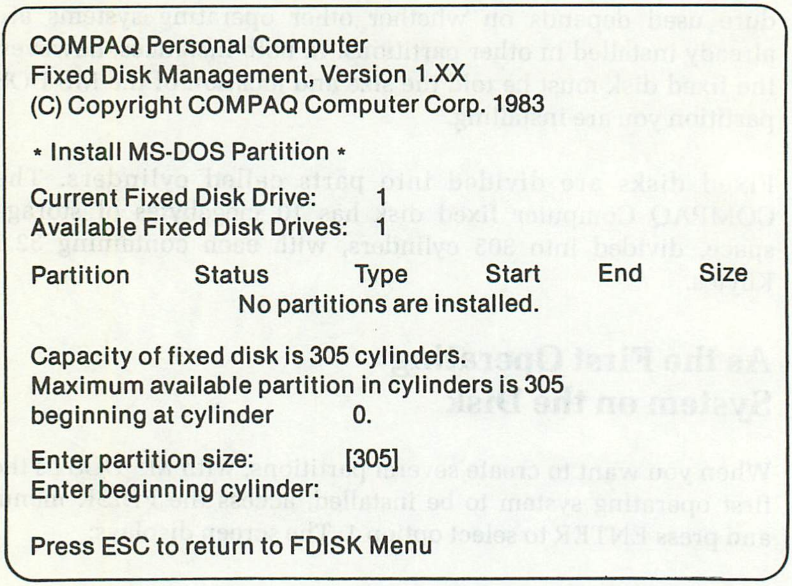
Fixed disks are divided into parts called cylinders. The COMPAQ Computer fixed disk has 10 megabytes of storage space, divided into 305 cylinders, with each containing 32.5 Kbytes.

As the First Operating System on the Disk

When you want to create several partitions, with MS-DOS as the first operating system to be installed, access the FDISK menu, and press ENTER to select option 1. The screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

 * Install MS-DOS Partition *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1


Do you wish to use the entire
fixed disk for MS-DOS (Y/N)?   [Y]




Press ESC to return to FDISK Menu
```

Since MS-DOS will occupy only one of several partitions, the answer is no. Enter N. The screen displays:



```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Install MS-DOS Partition *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Partition     Status      Type      Start     End     Size
                No partitions are installed.

Capacity of fixed disk is 305 cylinders.
Maximum available partition in cylinders is 305
beginning at cylinder          0.

Enter partition size:          [305]
Enter beginning cylinder:

Press ESC to return to FDISK Menu
```

The default value shows the largest area available on the fixed disk that can be used for a partition. Enter a number indicating the partition size you want to use for MS-DOS.

After the partition size is accepted, the brackets move to the next line and the screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Install MS-DOS Partition *

Current Fixed Disk Drive:        1
Available Fixed Disk Drives:     1

Partition      Status      Type      Start     End     Size
                   No partitions are installed.

Capacity of fixed disk is 305 cylinders.
Maximum available partition in cylinders is 305
beginning at cylinder             0.

Enter partition size:             305
Enter beginning cylinder:       [    0]

Press ESC to return to FDISK Menu
```

The default value shows the starting cylinder of the largest area that can be used for a partition. Enter a number indicating the first cylinder of the area that you want to become the MS-DOS partition.

After the computer accepts the cylinder number, the screen displays the new partition information. The following screen depicts a 40-cylinder partition that starts at cylinder 0.

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Install MS-DOS Partition *

Current Fixed Disk Drive:         1
Available Fixed Disk Drives:      1

Partition       Status       Type       Start    End    Size
    1          inactive     MS-DOS        0       39     40

Capacity of fixed disk is 305 cylinders.

Press ESC to return to FDISK Menu    [ ]
```

The brackets and the cursor move to the bottom of the screen. Press ESC to reach the FDISK menu.

If you want to start MS-DOS from your fixed disk, you must change it to an active partition. Proceed to Section 2-4, MODIFYING THE ACTIVE PARTITION.

The FDISK menu is displayed. Press ESC to return to MS-DOS. The screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Reset System *

Current Fixed Disk Drive:         1
Available Fixed Disk Drives:      1

An MS-DOS partition has been changed.
A system reset will occur to update MS-DOS.

Insert MS-DOS diskette in drive A.
Press ENTER when ready:    [  ]

Press ESC to return to FDISK Menu
```

When you press ENTER, a system reset occurs, and the screen displays the A> prompt.

Now you need to format the partition. Proceed to Section 2-3, FORMATTING A PARTITION.

With Other Operating Systems Already on the Disk

When you are creating an MS-DOS partition after other partitions containing different operating systems have already been created, access the FDISK menu and press ENTER to select option 1. The screen displays something similar to this:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Install MS-DOS Partition *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Partition     Status      Type      Start     End     Size
    1         ACTIVE      other         0      39       40
    2         inactive    other        40      99       50
    3         inactive    other       100     199      100

Capacity of fixed disk is 305 cylinders.
Maximum available partition in cylinders is 105
beginning at cylinder 200.

Enter partition size:          [105]
Enter beginning cylinder:

Press ESC to return to FDISK Menu
```

If the desired partition size is 75, for example, enter 75. If the desired partition size is 105, the default value, simply press ENTER. After you have entered the partition size, the bracket moves to the next line, and the screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983
 * Install MS-DOS Partition *
Current Fixed Disk Drive:        1
Available Fixed Disk Drives:     1
Partition       Status        Type       Start      End     Size
    1           ACTIVE        other          0       39       40
    2           inactive      other         40       99       50
    3           inactive      other        100      199      100
Capacity of fixed disk is 305 cylinders.
Maximum available partition in cylinders is 105
beginning at cylinder 200.
Enter partition size:             105
Enter beginning cylinder:        [200]
Press ESC to return to FDISK Menu
```

In this example, the appropriate number to enter (the next available starting cylinder) is 200. Since 200 is the default value, simply press ENTER.

After you enter the starting cylinder number, the screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983
 * Install MS-DOS Partition *
Current Fixed Disk Drive:        1
Available Fixed Disk Drives:     1

Partition       Status        Type       Start      End     Size
    1           ACTIVE        other          0       39       40
    2           inactive      other         40       99       60
    3           inactive      other        100      199      100
    4           inactive     MS-DOS        200      274       75

Capacity of fixed disk is 305 cylinders.

Press ESC to return to FDISK Menu    [  ]
```

The brackets and cursor move to the bottom of the screen. Press ESC to return to the FDISK menu.

If you want to start MS-DOS from your fixed disk, you must make it an active partition. Proceed to Section 2-4, MODIFYING THE ACTIVE PARTITION.

The FDISK menu is displayed. Press ESC to return to MS-DOS. The screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Reset System *

Current Fixed Disk Drive:       1
Available Fixed Disk Drives:    1

An MS-DOS partition has been changed.
A system reset will occur to update MS-DOS.

Insert MS-DOS diskette in drive A.
Press ENTER when ready:    [   ]


Press ESC to return to FDISK Menu
```

When you press ENTER, a system reset occurs and the screen displays the A> prompt.

Now you need to format the partition. Proceed to Section 2-3, FORMATTING A PARTITION.

2-3. FORMATTING A PARTITION

You must format your MS-DOS partition before you can use your fixed disk, regardless of whether the MS-DOS partition is installed as the entire disk, or as one of several partitions on the disk.

After you install an MS-DOS partition, the screen displays the A> prompt. Enter:

```
FORMAT C:/S/V
```

The /S sets up the system to load from the fixed disk by transferring the system files from the diskette to the MS-DOS partition on the fixed disk.

The /V allows you to assign a volume label. Volume label is the name for the contents of a diskette or a partition on a fixed disk. The volume label can be assigned only when you format the diskette or the partition. It is a good idea to always assign a volume label. This helps to further identify your diskettes, partitions, and files.

The screen now displays something like this:

```
A>FORMAT C:

Press any key to begin formatting C:
```

Once you strike a key, the screen displays:

```
Warning! The data in drive C will be destroyed.

Do you want to continue the format (Y/N)?    [N]
```

No (N) is the default response. If you enter N, the formatting procedure stops. If you enter Y, all existing data in the partition will be erased.

If you are formatting the partition on your fixed disk for the first time, there is no data to lose, so enter Y to continue formatting the partition. The screen then displays:

```
Formatting . . .
```

After several minutes, the screen displays:

```
Formatting . . . Format complete

System transferred
```

Then the screen displays the following message:

```
Volume label (11 characters, ENTER for none)?
```

You may enter up to 11 characters for the volume label, or press ENTER if you do not want to assign a volume label.

NOTE: You must assign the volume label now. It cannot be added later.

When the formatting is complete, the screen displays disk space statistics similar to this:

```
10592256 bytes total disk space

    41472 bytes used by system

10550784 bytes available on disk
```

The screen then displays the A> prompt. In response, enter:

```
COPY *.* C:
```

This command copies all the files that are on the MS-DOS diskette, including the internal and external commands, onto the fixed disk. This allows you to access all the commands from the fixed disk.

Remove the MS-DOS diskette from Drive A:, open the disk drive door(s), and do a system reset (^ALT+DEL). You must remove the diskette or open the drive door(s) for the system to load from the fixed disk. The computer cycles through its diagnostics and displays the C> prompt. Your fixed disk is now ready to accept commands.

2-4. MODIFYING THE ACTIVE PARTITION

This option is used when you want to change the operating system your computer loads from.

To change an inactive partition to an active partition, access the FDISK menu, and enter 2. The screen displays something similar to this:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Modify Active Partition *

Current Fixed Disk Drive:   1
Available Fixed Disk Drives:   1

Partition      Status      Type      Start     End     Size
    1         inactive    MS-DOS       0        39      40
    2         ACTIVE       other     200       299     100

Capacity of fixed disk is 305 cylinders.
The current active partition is 2.

Enter partition number to activate:   [   ]

Press ESC to return to FDISK Menu
```

To make the MS-DOS partition active, enter 1. The screen displays the new active partition. For example, if you entered 1, the previous screen now changes to:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983
 * Modify Active Partition *

Current Fixed Disk Drive:       1
Available Fixed Disk Drives:    1

Partition       Status       Type      Start    End    Size
    1           ACTIVE      MS-DOS       0       39     40
    2           inactive     other      200     299    100

Capacity of fixed disk is 305 cylinders.
The current active partition is 1.

Enter partition number to activate:    1

Press ESC to return to FDISK Menu    [  ]
```

The brackets and the cursor move to the bottom of the screen. Press ESC to return to the FDISK menu. Press ESC again to return to MS-DOS.

2-5. REMOVING AN MS-DOS PARTITION

You usually remove an MS-DOS partition when you want to recreate it in a different size or location.

CAUTION

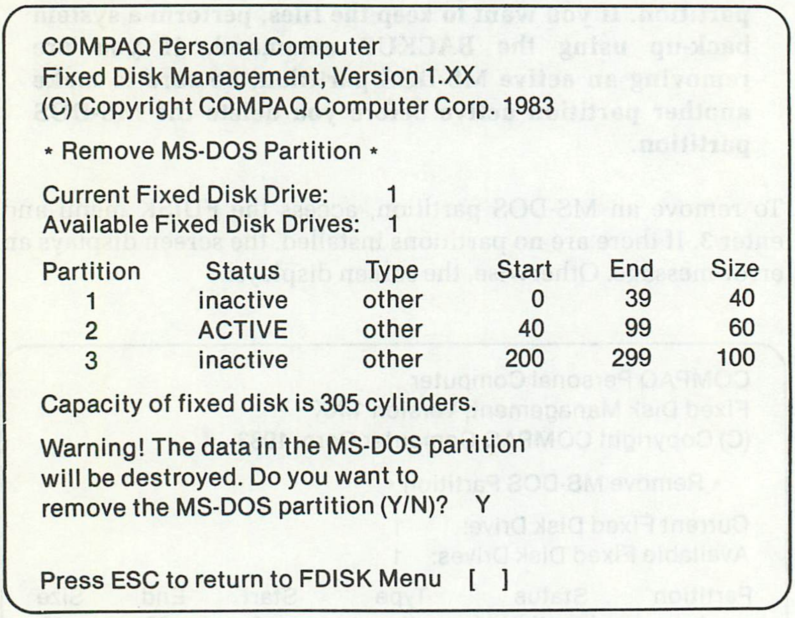
Removing an MS-DOS partition will erase all files in the partition. If you want to keep the files, perform a system back-up using the BACKUP command. If you are removing an active MS-DOS partition, be sure to make another partition active before you delete the MS-DOS partition.

To remove an MS-DOS partition, access the FDISK menu and enter 3. If there are no partitions installed, the screen displays an error message. Otherwise, the screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

   * Remove MS-DOS Partition *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Partition     Status      Type      Start    End    Size
    1        inactive     other        0      39      40
    2        inactive     other       40      99      60
    3        inactive    MS-DOS      100     199     100
    4        ACTIVE       other      200     299     100
Capacity of fixed disk is 305 cylinders.

Warning! The data in the MS-DOS partition
will be destroyed. Do you want to
remove the MS-DOS partition (Y/N)?    [N]

Press ESC to return to FDISK Menu
```

If you don't want to remove the MS-DOS partition, press ENTER (for the default response no) or the ESC key to return to the FDISK menu.

If you want to remove the MS-DOS partition, enter Y. The system deletes the partition and displays a revised screen. For example, after the deletion, the screen above will not show the MS-DOS partition or its information. The cursor and the bracket move to the bottom of the screen, and the screen displays:



```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

 * Remove MS-DOS Partition *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Partition      Status      Type      Start    End     Size
    1         inactive     other        0      39       40
    2         ACTIVE       other       40      99       60
    3         inactive     other      200     299      100

Capacity of fixed disk is 305 cylinders.

Warning! The data in the MS-DOS partition
will be destroyed. Do you want to
remove the MS-DOS partition (Y/N)?   Y


Press ESC to return to FDISK Menu    [  ]
```

Press ESC to return to the FDISK menu. If you press ESC again to exit from FDISK, the screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* Reset System *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1


An MS-DOS partition has been changed.
A system reset will occur to update MS-DOS.


Insert MS-DOS diskette in drive A.
Press ENTER when ready:    [ ]



Press ESC to return to FDISK Menu
```

When you press ENTER, a system reset occurs and the screen displays the A> prompt.

2-6. LISTING PARTITION DATA

To list partition information, access the FDISK menu and enter 4. If no partitions are installed, the screen displays an error message. Otherwise, the screen displays something similar to this:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* List Partition Information *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Partition      Status       Type       Start    End     Size
    1          ACTIVE       MS-DOS         0     39       40
    2          inactive     other         40     99       60
    3          inactive     other        100    199      100
    4          inactive     other        200    299      100

Capacity of fixed disk is 305 cylinders.

Press ESC to return to FDISK Menu    [  ]
```

Press ESC to return to the FDISK menu.

2-7. SELECTING ANOTHER FIXED DISK

If you have more than one fixed disk installed, option 5 appears on the FDISK menu. To select another fixed disk, enter 5. The screen displays:

```
COMPAQ Personal Computer
Fixed Disk Management, Version 1.XX
(C) Copyright COMPAQ Computer Corp. 1983

* FDISK Menu *

Current Fixed Disk Drive:      2
Available Fixed Disk Drives:   2

Choose one of these options:

   1. Install MS-DOS Partition
   2. Modify Active Partition
   3. Remove MS-DOS Partition
   4. List Partition Information
   5. Select Other Fixed Disk Drive

Enter your choice:   [1]
```

In this sample screen, after entering 5, the current fixed disk drive changed from 1 to 2.

CHAPTER 3

TREE-STRUCTURED DIRECTORIES

3-1. THE DIRECTORY STRUCTURE 3-1
3-2. PATHING. 3-3
Pathnames 3-3
Working Directory Notations 3-4
3-3. DIRECTORY COMMANDS 3-6
Creating a Subdirectory: The MKDIR Command 3-6
Listing Files in a Working Directory: The DIR Command 3-7
Changing a Working Directory: The CHDIR Command. . 3-8
Deleting a Directory: The RMDIR Command 3-9
Displaying the Directory Structure: The TREE Command. 3-9

3-1. THE DIRECTORY STRUCTURE

Each disk has a directory that gives the name, location, size, creation date, and last revision date of each file on the disk. In previous versions of DOS, which did not support a fixed disk, a simple directory structure was sufficient. With the addition of a fixed disk, which can store many more files than a diskette, a more sophisticated method of organizing files is required to enable you to efficiently store and retrieve data. MS-DOS provides the capability to organize files into several levels of directories, which allows you to group your files (by content or user, for example).

This method of organizing files into directories is called hierarchical directory structuring, or tree structuring. The term tree structuring is used because the directories are organized somewhat like an inverted tree, with a root, branches, and leaves. The root directory is the first level of the directory structure. The branches are the subdirectories, which can contain data files or other subdirectories, and the leaves are the files. The "tree" grows as new directories are added for groups of files or new users. An example of a tree-structured directory is shown in Figure 3-1.

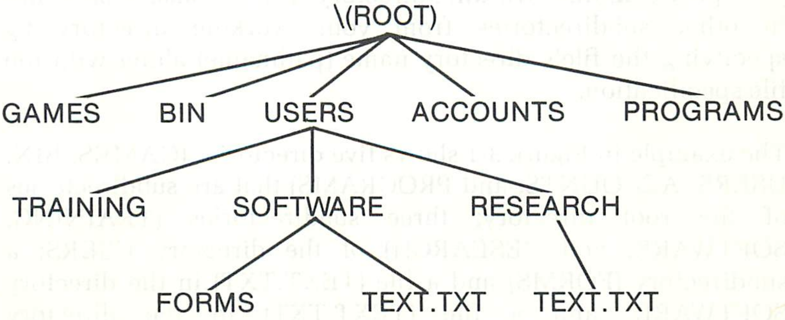


Figure 3-1. A Tree-Structured Directory

The **root directory** is the main directory on a disk, and is designated by a backslash (\). The root directory is created when the disk is formatted, and cannot be deleted. It is the directory in which you start when you first access the disk. The maximum number of entries in a root directory depends on the number of bytes formatted by MS-DOS. For example, the root directory on a formatted, double-sided diskette may contain up to 112 entries. When the MS-DOS partition has been created using the entire fixed disk, the root directory may contain up to 512 entries.

A **subdirectory** is any directory other than the root directory. A subdirectory is actually a file that contains data files and other subdirectory files. Therefore, the rules for naming a file also apply to naming a subdirectory. Since a subdirectory has no size restriction, the number of files or other subdirectories it can contain is limited only by the amount of space available on the disk.

In this manual, the default directory is called the **working directory.** It is sometimes referred to as the current directory. The working directory is the directory the computer accesses when you specify a file without specifying a directory (the computer defaults to the working directory). By using MS-DOS commands, you can choose your working directory. When you turn the computer on, the root directory is the working directory, until you specify another working directory. You can also access files in other subdirectories from your working directory by specifying the file's directory name (pathname) along with the file specification.

The example in Figure 3-1 shows five directories (GAMES, BIN, USERS, ACCOUNTS, and PROGRAMS) that are subdirectories of the root directory; three subdirectories (TRAINING, SOFTWARE, and RESEARCH) of the directory USERS; a subdirectory (FORMS) and a file (TEXT.TXT) in the directory SOFTWARE; and a file (TEXT.TXT) in the directory RESEARCH. Because SOFTWARE and RESEARCH are different subdirectories, they may both have a file named TEXT.TXT.

3-2. PATHING

Before the computer can find a file in a directory, it has to know where to look to locate it. As previously discussed, files are given names consisting of a series of characters, preceded by an optional drive designator and followed by an extension. Before MS-DOS can locate a file in a tree-structured directory, you must tell it in which subdirectory the file is located. This is done by specifying a pathname for the file.

PATHNAMES

A pathname is a sequence of subdirectory names, followed by a file specification, with a backslash between each as a delimiter. Each full pathname can have as many as 63 characters. The syntax for pathnames is:

[*d:*] [*directory*]\[*directory...*]\[*filespec*]

If a pathname begins with a backslash, MS-DOS begins its file search in the root directory. Otherwise, MS-DOS starts looking in the working directory. For example, the pathname for Software's TEXT.TXT file is \USER\SOFTWARE\TEXT.TXT.

Some sample pathnames are shown below:

\ACCOUNTS	A subdirectory under the root directory that contains accounting files.
\USERS\RESEARCH	A subdirectory belonging to the user department Research.
\USERS\SOFTWARE\FORMS\1A	A file named 1A in a subdirectory named FORMS belonging to the user department Software.
TEXT.TXT	Name of a file in the working directory.

WORKING DIRECTORY NOTATIONS

To help you access files faster, MS-DOS provides special shorthand notations. These notations are present in every subdirectory.

.	represents the name of a subdirectory in all hierarchical directories. This entry is automatically created when a directory is created.
..	represents the parent directory of the working directory. A parent directory is the directory one level above the working directory.

For example, if you are in the subdirectory FORMS (shown in the example of the tree-structured directory) and enter:

```
DIR ..
```

the screen displays all the files in the subdirectory SOFTWARE.

If you are in the subdirectory FORMS and enter:

```
DIR ..\..
```

the screen displays all the files in the subdirectory USERS.

NOTE: If you use the .. notation in the pathname, MS-DOS will begin its file search in the parent directory. For example, if you are in the subdirectory SOFTWARE and enter:

```
..
```

the system searches in the subdirectory USERS for the file.

3-3. DIRECTORY COMMANDS

MS-DOS uses several commands to list, create, locate, change, and delete directories. These are listed below, and detailed in the following sections.

Command:	**Syntax:**	**Function:**
MKDIR	MKDIR [*d:*] [*pathname*] or MD [*d:*] [*pathname*]	Creates a new subdirectory.
DIR	DIR [*d:*] [*pathname*]	Lists all the files in the pathname.
CHDIR	CHDIR [*d:*] [*pathname*] or CD [*d:*] [*pathname*]	Changes or displays the working directory.
RMDIR	RMDIR [*d:*] [*pathname*] or RD [*d:*] [*pathname*]	Deletes (removes) a subdirectory.
TREE	TREE [*d:*] [*pathname*] [/F]	Displays the tree structure of a directory and (optionally) lists the files in each subdirectory.

CREATING A SUBDIRECTORY: THE MKDIR COMMAND

Use the MKDIR command to create subdirectories anywhere in the tree-structured directory or in your working directory. When you are in the root directory, create a subdirectory by entering the MKDIR command, a backslash (denoting the root directory), and the name of the subdirectory you are creating. For example, enter:

```
MKDIR \USERS
```

creates a subdirectory named USERS under the root directory.

To create a subdirectory while in your working directory, enter MKDIR and the name of the subdirectory you want to create. For example, entering:

```
MKDIR RESEARCH
```

creates a new subdirectory named RESEARCH under your working subdirectory.

To create a directory elsewhere in the tree-structured directory, enter MKDIR and a pathname. For example, entering:

```
MKDIR \ACCOUNTS\BILLS
```

creates a subdirectory named BILLS under the subdirectory ACCOUNTS.

MS-DOS automatically creates the . and .. entries in the new directory.

LISTING FILES IN A WORKING DIRECTORY: THE DIR COMMAND

The DIR command is used to list all the files in your working directory. When you enter DIR, a drive designator, and a directory name, the computer displays a directory that shows the file specifications and their extensions, the word DIR (if the file is a subdirectory) or the file size (if it is a file), the number of bytes free, and the date and time that each file was last modified.

For example:

```
A> DIR B:NEWDIR\SUBDIR\USERS\RESEARCH

Volume in drive B is PLAYDISK
Directory of B: \NEWDIR\SUBDIR\USERS\RESEARCH

.                   <DIR>        1-01-80  12:04a
..                  <DIR>        1-01-80  12:04a
AFILE                      105   1-01-80  12:07a
TEXT       BAK               3   1-01-80  12:14a
TEXT                        84   1-01-80  12:17a
MEMOS               <DIR>        1-01-80  12:31a

         6 File(s)     346112 bytes free
```

For more details on the DIR command, see Chapter 4, MS-DOS COMMANDS.

CHANGING A WORKING DIRECTORY: THE CHDIR COMMAND

This command allows you to travel to any subdirectory in the tree-structured directory by using CHDIR and a pathname.

To change from your working directory to another directory, enter CHDIR and a pathname. For example, to change a working directory from \USERS\SERVICE to USERS, enter:

```
CHDIR \USERS
```

Entering CHDIR .. always puts you in the parent directory of your working directory.

DELETING A DIRECTORY: THE RMDIR COMMAND

Before you can delete a subdirectory, you must first delete the files in that subdirectory. Remember that the . and .. entries cannot be removed because they are created by MS-DOS as part of the hierarchical directory structure.

For example, to delete the files in the \SUBDIR\USERS\SOFTWARE directory, enter:

```
DEL \SUBDIR\USERS\SOFTWARE *.*
or
DEL \SUBDIR\USERS\SOFTWARE
```

When you specify *.*, the message "Are you sure (Y/N)?" appears. If you do not specify a file, the system defaults to *.*.

Once these files are deleted (except for the . and .. entries), delete the subdirectory by specifying its pathname. In this case, you enter:

```
RMDIR \SUBDIR\USERS\SOFTWARE
```

The entire subdirectory is deleted.

DISPLAYING THE DIRECTORY STRUCTURE: THE TREE COMMAND

The TREE command lists all subdirectories on the disk by path. For example, when you enter:

```
TREE A:
```

the system lists all subdirectories on the disk in Drive A: by path.

If you specify the /F switch, the names of the files in the subdirectories are also displayed. In this example, when you enter:

```
TREE A:/F
```

the following list appears:

```
DIRECTORY PATH LISTING FOR VOLUME PLAYDISK

Path:  \NEWDIR

Sub-directories:  SUBDIR

Files:            None

Path:  \NEWDIR\SUBDIR

Sub-directories:  USERS

Files:            None

Path:  \NEWDIR\SUBDIR\USERS

Sub-directories:  TRAINING
                  RESEARCH
                  SOFTWARE

Files:            None

Path:  NEWDIR\SUBDIR\USERS\TRAINING

Sub-directories:  None

Files:            None

Path:  NEWDIR\SUBDIR\USERS\RESEARCH
```

Sub-directories: MEMOS

Files: AFILE
TEXT BAK
TEXT

Path: \NEWDIR\SUBDIR\USERS\RESEARCH\MEMOS

Sub-directories: None

Files: INDEXER

Path: \NEWDIR\SUBDIR\USERS\SOFTWARE

Sub-directories: None

Files: TEXT.2

Graphically, these directories and the paths to them appear on a tree-structured directory as shown in Figure 3-2.

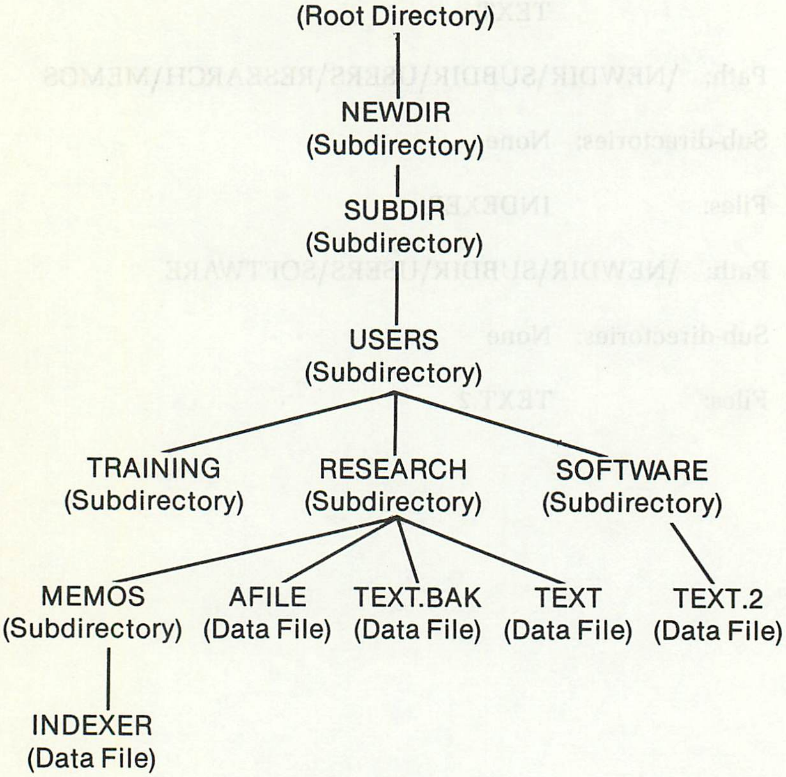


Figure 3-2. Directories on the Disk in Drive A:.

CHAPTER 4

MS-DOS COMMANDS

4-1. INTRODUCTION 4-1
4-2. TYPES OF MS-DOS COMMANDS 4-2
4-3. COMMAND OPTIONS 4-4
4-4. RULES FOR USING MS-DOS COMMANDS 4-5
4-5. BATCH PROCESSING 4-7
  The AUTOEXE.BAT File. 4-7
  Creating a Batch File With Replaceable Parameters. 4-9
  Executing a Batch File 4-10
  Batch Commands. 4-11
4-6. INPUT AND OUTPUT 4-13
  Redirecting Output 4-13
  Redirecting Input. 4-14
  Filters 4-14
  Command Piping 4-15
4-7. HOW TO CONFIGURE YOUR SYSTEM 4-16
  Configuration Commands 4-16
4-8. COMMANDS. 4-18

4-1. INTRODUCTION

Commands provide the interface between you and the operating system. When you enter commands, you are telling the system to perform certain tasks, such as copy, compare, display, delete, and rename files; copy, compare, and format disks; execute system programs; analyze and list directories; and set input and output options.

4-2. TYPES OF MS-DOS COMMANDS

MS-DOS commands are either internal or external. Internal commands reside in memory after the operating system loads. They execute immediately when entered. They do not, however, appear on the directory listing.

External commands reside on the disk as program files, so they must be read from the disk before they can execute. External files, therefore, can be created and added to the system. If the command is not on the disk that is in the drive you are using, MS-DOS cannot locate and execute it (see the PATH command).

File specifications with .COM or .EXE extensions are external commands. Do not include the extensions when you type these commands.

The internal and external commands on the MS-DOS diskette are listed below. Each of these commands is detailed in this chapter.

Internal Commands:

BREAK
BUFFERS
CHDIR
CLS
COPY
CTTY
DATE
DEL
DEVICE
DIR
ECHO
ERASE
FILES
FOR
GOTO
IF
MKDIR
PATH
PAUSE
PROMPT
REM
REN
RMDIR
SET
SHELL
SHIFT
TIME
TYPE
VER
VERIFY
VOL

External Commands and Device Drivers:

ANSI.SYS
ASSIGN
BACKUP
CHKDSK
COMP
DISKCOMP
DISKCOPY
EXE2BIN
FDISK
FIND
FORMAT
GRAPHICS
MODE
MORE
PRINT
RECOVER
RESTORE
SORT
SYS
TREE

The internal commands used to configure your system are:

BREAK
BUFFERS
DEVICE
FILES
SHELL

4-3. COMMAND OPTIONS

To supply additional information to the system, include parameters in the command. A parameter is a variable that may be assigned different values during a program run. If a parameter is not included, MS-DOS sometimes provides a default value. Default values for each command are included in the command descriptions in Section 4-8, COMMANDS.

The syntax for all MS-DOS commands is:

```
COMMAND   [parameter]
```

The parameters can be:

- Drive designators.
- Pathnames that include a series of directory names and a file specification.
- File specifications that include filenames and extensions.
- Switches that designate a course of action for the command. A switch is designated by a slash (/) followed by a letter or number.
- Arguments that usually require a choice, such as ON or OFF. An argument is a type of variable whose value is not a direct function of another variable.

4-4. RULES FOR USING MS-DOS COMMANDS

Rules that must be followed when using MS-DOS commands appear below:

- Type commands in upper or lower case (or a combination of both) as indicated.

- Separate commands from parameters with delimiters, such as spaces, commas, slashes, or semicolons. In this book, a space is most often used as a delimiter.

- Do not separate file specifications with delimiters. The colon between the drive specifier and the file specification, the backslashes in the pathname, and the period between the filename and the extension serve as delimiters.

- Include the file's extension, if it has one, in the file specification.

- Press any key on the keyboard whenever you are prompted to "Strike any key". The SPACEBAR or the ENTER key is typically used.

- To abort commands that are in progress, press ^BREAK.

- To enter typed commands, press ENTER.

- Do not use wild card characters or device names in command names.

- Press ^NUM LOCK when you want to suspend scrolling on the screen. Press any other key to continue the screen display.

- Use MS-DOS editing and function keys when entering commands.

- The command processor prompt is the default drive designation letter plus a greater-than character (for example, A>), unless it has been changed with the PROMPT command.

- Disk drives are sometimes designated as source or target drives. Information is always transferred from the source drive to the target drive.

4-5. BATCH PROCESSING

With MS-DOS, you can put a sequence of commands into a batch file to save you from typing identical commands repeatedly. With batch processing, the system executes the entire sequence of commands as if you had entered each one individually. When using batch processing, you must include the .BAT extension when you name a file, but exclude the extension when typing the file specification as a command.

REM and PAUSE are two of the commands especially created for use in batch files. REM allows you to use remarks and comments in the batch files without having them interpreted as commands. PAUSE allows you to suspend the batch process at a given point. A sample batch file is:

```
REM          This file checks new disks
REM          Its name is NEWDISK.BAT
PAUSE        Insert new disk in Drive B:
FORMAT B:
DIR B:
CHKDSK B:
```

To execute this batch file, simply enter the filename NEWDISK without the extension. NEWDISK then becomes an MS-DOS command. Each command in the batch file executes as if it were entered individually.

THE AUTOEXEC.BAT FILE

An AUTOEXEC.BAT file contains instructions that are automatically executed when you turn the system on. When the MS-DOS program is loaded, the COMMAND.COM file searches the disk for an AUTOEXEC.BAT file. If one is found, the computer acts on the instructions it contains and bypasses the date and time prompts, unless they have been created as part of the AUTOEXEC.BAT file. If there is no AUTOEXEC.BAT file, the system displays the date and time prompts.

For example, to automatically load BASIC and execute the program called PAYABLES, enter:

```
COPY CON: AUTOEXEC.BAT
```

This statement tells MS-DOS to copy input from the keyboard to the AUTOEXEC.BAT file.

Now enter:

```
BASIC PAYABLES
```

This statement is inserted in the AUTOEXEC.BAT file. It tells MS-DOS to load BASIC and run the PAYABLES program.

Now press F6 and ENTER. The screen displays:

```
1 File(s) copied

A>
```

This indicates the termination of the COPY command from CON: (by pressing F6 and ENTER). The PAYABLES program now executes automatically when you load MS-DOS.

NOTE: Include the DATE and TIME commands in the AUTOEXEC.BAT file, if you want the computer to use the correct date and time.

CREATING A BATCH FILE WITH REPLACEABLE PARAMETERS

Creating a batch file that has replaceable parameters allows you to have it run with different sets of data that are housed in various MS-DOS files. Parameters named %0 through %9 are available to take on specified values when batch files execute. If more than 10 parameters are needed, use the SHIFT command (see SHIFT Command).

To create a batch file named MYFILE with three replaceable parameters, type the following lines, and press ENTER after each line. For example:

```
COPY CON MYFILE.BAT
COPY %1.MAC %2.MAC
TYPE %2.PRN
TYPE %0.BAT
```

Press F6 and ENTER. The screen displays:

```
1 File(s) copied

A>
```

MYFILE.BAT now includes three commands: COPY, TYPE, and TYPE. In a batch file with replaceable parameters, the %0 is always replaced by the drive designator, if given, and the batch file specification. Parameters %1 and %2 are replaced sequentially by the parameters you supply when you execute the file.

To define replaceable parameters with names instead of numbers, use the SET command.

Use SET prior to executing the batch command, or within the batch file prior to executing a command. When defining the parameter, type a percent sign before and after the parameter name.

For example, if you have a batch file:

```
REM Batch command
TYPE %File%
TYPE \SUBD\%File%BAT
```

and you enter:

```
SET FILE=TEST
```

Upon execution of the batch command, TEST replaces %FILE%, and the screen displays:

```
REM    BATCH COMMAND

TYPE   TEST

TYPE   \SUBD\TEST.BAT
```

If you need to use the percent character (%) as part of a file specification within a batch file, type the % twice. For example, to specify the file ABC%.EXE, type ABC%%.EXE in the batch file.

EXECUTING A BATCH FILE

To execute a batch file and specify the parameters that will replace the dummy parameters, enter the name of the batch file without the extension and the parameters you want MS-DOS to use. For example, consider the following batch file:

```
REM START TEST BATCH FILE
COPY %1 %2
TYPE %2.PRN
TYPE %0.BAT
```

To execute the batch file named ASMFILE.BAT and assign the parameters, enter the batch file specification and the parameters that you want to be sequentially entered. For example:

```
ASMFILE A:MYFILE B:MYPROG
```

Since the drive designator (if one is used) and the batch file specification are always substituted for the replaceable parameter %0, A:MYFILE will replace %1, and B:MYPROG replaces %2. The result will appear as if you had entered each command from the computer with the indicated parameter, as follows:

```
REM START TEST BATCH FILE
COPY A:MYFILE.ASM B:MYPROG.ASM
TYPE B:MYPROG.PRN
TYPE A:ASMFILE.BAT
```

BATCH COMMANDS

There are five batch processing commands: ECHO, FOR, GOTO, IF, and SHIFT. These internal commands add flexibility and power to the batch files you write. A description of each of these commands follows, and is detailed in Section 4-8, COMMANDS.

Name:	**Function:**	**Syntax:**
ECHO	Turns batch ECHO on and off to prevent or cause batch commands to be displayed on the screen.	ECHO [ON|OFF|*message*]

Continued

Name:	**Function:**	**Syntax:**
FOR	Executes a command repeatedly for each value of the variable listed in *set*.	FOR %%*c* IN(*set*) DO *command*
GOTO	Continues processing at the specified label in the batch file.	GOTO *label*
IF	Allows conditional execution of commands.	IF [NOT] *condition* *command*
PAUSE	Suspends execution of the batch file	PAUSE [*comment*]
REMARK	Displays a remark in a batch file during batch file execution.	REM [*comment*]
SHIFT	Allows more than 10 parameters in batch file processing.	SHIFT

4-6. INPUT AND OUTPUT

MS-DOS assumes that input comes from the keyboard and output goes to the screen. The flow of command input and output, however, can be redirected. By entering the proper command, input can come from a file, and output can go to a file or an optional line printer. Data can also be redirected so that the output from one line command becomes the input to another.

REDIRECTING OUTPUT

The output from most commands goes to the screen. By including a greater-than character (>) in the command, you can send the output to a file. For example, when the DIR command is entered, the screen displays a list of all the files on the disk in the specified or default drive. To send this directory listing to another file, type the command, the greater-than character (>), the drive designator, and the name of the file you are sending it to. For example, enter:

```
DIR >B:MYFILES
```

In this example, MS-DOS creates the MYFILES file, if it doesn't already exist. If it already exists, it replaces the data in MYFILES with the new data. If the information is to be added to the end of the file rather than replace the data in the file, use two greater-than characters. For example:

```
DIR >>B:MYFILES
```

tells MS-DOS to send the directory listing to the end of the file named MYFILES on the diskette in Drive B:.

If a drive designator isn't specified, the system creates the file on the disk in the default drive.

REDIRECTING INPUT

By including a less-than character (<) in the command, MS-DOS takes the input from a file rather than from the screen. For example, if you enter:

```
SORT <NAMES >LIST1
```

the system sorts the file NAMES, and sends the sorted output to replace the data in a file called LIST1. If you want to add the information to the end of the LIST1 file, enter:

```
SORT <NAMES >>LIST1
```

FILTERS

Filters are are commands that read and transform input. Filters can replace many specific commands. Descriptions of MS-DOS filters follow, and are detailed in Section 4-8, COMMANDS.

Name:	**Function:**	**Syntax:**
FIND	Searches for specific strings of text in a file.	FIND [/V] [/C] [/N] "*string*" [[*d:*] [*pathname*] *filespec*...]
MORE	Displays terminal output one screen at a time.	MORE
SORT	Sorts text.	SORT [/R][/ + *n*]

COMMAND PIPING

Piping is the process used to issue more than one command simultaneously. When piping, separate each command with a broken bar (¦). The system performs the function on the left side of the bar, and then sends it to the right side for processing. If more than two commands are used, the system performs the tasks as the commands progress to the right. For example, entering:

```
DIR¦SORT>DIREC.FIL
```

tells MS-DOS to sort the directory and send the sorted list to replace the information in a file named DIREC.FIL. If DIREC.FIL doesn't exist, MS-DOS creates it on the disk in the default drive, since a drive designation was not specified. To send this information to a file on a diskette in Drive B:, enter:

```
DIR¦SORT>B:DIREC.FIL
```

To send this alphabetized directory to the end of the DIREC.FIL file, enter:

```
DIR¦SORT ≫B.DIREC.FIL
```

To tell MS-DOS to sort the directory and display it one screen at a time, enter:

```
DIR¦SORT¦MORE
```

MORE appears at the bottom of the screen when there is more data to display.

NOTE: Redirections of output, pipes, and filters work only if the applications program you are in uses the MS-DOS function calls.

When piping is used, temporarary files are created in the root directory of the default drive. These files are named %PIPEX.$$$. If the diskette in the default drive is write-protected, a write-protect error occurs.

4-7. HOW TO CONFIGURE YOUR SYSTEM

In MS-DOS, there is a file named CONFIG.SYS. that can be set up to contain input configuration commands that are used to perform special functions.

When MS-DOS is loaded, COMMAND.COM looks in the root directory for the CONFIG.SYS file. It it is not there, the system goes to the date and time prompts.

CONFIGURATION COMMANDS

Five commands (BREAK, BUFFERS, DEVICE, FILES, and SHELL) have been especially created for configuring your system. These commands, with their syntax and functions, follow. See Section 4-8, COMMANDS, for a complete description of each command.

Command:	**Syntax:**	**Function:**
BREAK	BREAK ON|OFF	Turns extended ^BREAK checking on and off.
BUFFERS	BUFFERS = xx	Indicates the number of disk buffers allocated in memory.
DEVICE	DEVICE = [*d*:] [*pathname*] *filespec*	Directs MS-DOS to additional files that contain device drivers for additional devices.

Command:	**Syntax:**	**Function:**
FILES	FILES = xx	Specifies the maximum number of files that can be opened at any time.
SHELL	SHELL = [*d*:] [*pathname*] *filespec*	Allows the use of a command processor other than COMMAND.COM to load the system.

4-8. COMMANDS

Listed on the following pages are detailed descriptions of the MS-DOS commands. These commands are arranged in alphabetical order. Each command section gives the command type; that is, whether the command is internal, external, or a utility, and where applicable, whether it is a batch command, a filter, or a configuration command. Each command section also includes the purpose of the command, its syntax, and a series of comments that usually include examples for using the command.

ANSI.SYS Device Driver

Type: Extended screen and keyboard device driver.

Purpose: To make available (via the CONFIG.SYS file) the options of moving the cursor, changing the screen display, and assigning specific key functions.

Syntax: (See individual control sequences)

Comments: The following are extended control sequences used in cursor positioning, screen display options, and key reassignments. These control sequences can only be used if the ANSI.SYS device driver has been installed (by using the command DEVICE = ANSI.SYS) in the CONFIG.SYS file.

Definitions of symbols used in the control sequence syntax are as follows:

ESC	Equals 1B hex in all control sequences. ESC stands for a unique character; it does not stand for the ESC key, or the separate letters E, S, and C.
#	Indicates a variable numeric value.
...	Indicates continuing variables as specified.
;	Separates variables and/or specifications. The semicolon must be included where specified.

Various letters and numbers are included in the control sequences. They must be entered exactly as indicated.

The syntax statement for each control sequence (left column), acronym for the function (right column), and brief description follow.

Positioning the Cursor

The following are the syntax statements for specifying various cursor positions.

Syntax:	Description:
ESC[#;#H	CUP - cursor position Sets the horizontal and vertical (x/y coordinates) positions of the cursor. The first variable can have a value of 1-25; the second, 1-80 (1-40 if you are in the 40 × 25 display mode). The default value for both variables is 1. If no value or 0 (NUL) is specified, the cursor returns to its home position at the top left corner of the screen.
ESC[#A	CUU - cursor up Moves the cursor up by the number of lines specified. The variable can have a value of 1-24, with a default of 1.
ESC[#B	CUD - cursor down Moves the cursor down by the number of lines specified. The variable can have a value of 1-24, with a default of 1.

Syntax:	**Description:**
ESC[#C	CUF - cursor forward Moves the cursor to the right by the number of columns specified. The variable can have a value of 1-79 (1-39), with a default of 1.
ESC[#D	CUB - cursor backward Moves the cursor to the left by the number of columns specified. The variable can have a value of 1-79 (1-39), with a default of 1.
ESC[#;#f	HVP - horizontal/vertical position Changes the horizontal and vertical position of the cursor. The first variable can have a value of 1-25; the second, 1-80 (1-40). The default for both variables is 1. This is the same as CUP.
ESC[6n	DSR - device status report Requests a cursor position status report. The device driver replies with the following: ESC[#;#R The first variable gives the vertical position of the cursor, and the second variable gives the horizontal position.

Continued

Continued

Syntax:	**Description:**
ESC[s	SCP - save cursor position Saves the cursor position for later recall via the RCP sequence.
ESC[u	RCP - restore cursor position Restores the cursor to the previously-saved position.
ESC[2J	ED - erase display Erases the entire display and positions the cursor at the home position.
ESC[k	EL - erase line Erases a line from the cursor position to the end of the line, including the cursor position.

Screen Display Options

The following are the syntax statements for the various screen display options.

Syntax:	**Description:**
ESC[#;...;#m	SGR - set graphics rendition Sets various screen modes (display attributes, and background and foreground characteristics). All subsequently-written characters will have those attributes until the next occurrence of an SGR sequence. The variables can be replaced with any combination of values from the following table:

Parameter:	**Description:**
0	All attributes off (default = white characters on black background)
1	High intensity (boldface) on
4	Underscore (underline) on (monochrome only)
5	Blinking (flashing) on
7	Inverse video (black-on-white) on
8	Invisible (black-on-black) display
30	Foreground black
31	Foreground red
32	Foreground green
33	Foreground yellow
34	Foreground blue
35	Foreground magenta
36	Foreground cyan
37	Foreground white
40	Background black
41	Background red
42	Background green
43	Background yellow
44	Background blue
45	Background magenta
46	Background cyan
47	Background white

Continued

Continued

ESC[=#h
or
ESC[=h
or
ESC[?#h

SM - set screen mode

Sets the operating mode of the screen display. The variable can be replaced with one of the values from the following table:

Parameter:	**Description:**
0	40 × 25 monochrome
1	40 × 25 color
2	80 × 25 monochrome
3	80 × 25 color
4	320 × 200 color
5	320 × 200 monochrome
6	640 × 200 monochrome
7	Enables wrap at end-of-line (typing past end-of-line causes automatic carriage return)

ESC[=#l
or
ESC[=l
or
ESC[?#l

RM - resets screen mode

Resets the operating mode of the screen display after the initial change. The variable can be replaced with one of the values from the following table:

Parameter:	**Description:**
0	40 × 25 monochrome
1	40 × 25 color
2	80 × 25 monochrome
3	80 × 25 color
4	320 × 200 color
5	320 × 200 monochrome
6	640 × 200 monochrome
7	Disables wrap at end-of-line (characters past end-of-line are thrown away)

ESC[#;#;...#p or ESC[“*string*”;p or ESC[#;“*string*”;#; #;“*string*”;#p	Reassigns the function of any key according to your specifications. The first variable is the character code for the key you want to reassign. If you want to use an extended character code (ECC), replace the first variable with 0 (NUL), and use the second variable for the ECC value. The string can be any phrase or command you want to use. The last variables are the character codes for the values (or characters) you are assigning to the key. The variables can be any combination of decimal numbers and/or strings. Refer to the ***BASIC VERSION 2 REFERENCE GUIDE*** for a list of ASCII characters and extended character codes.

Examples:

1) F10 key reassigned to execute a FORMAT B: command:

```
ESC [0;68;“FORMAT B:”;13p
```

(0;68 = extended character code for F10 key; FORMAT B: = command to execute; 13 = decimal ASCII character code for ENTER)

2) @ (“at” sign) key reassigned to √ (square root sign):

```
ESC [64;251p
```

(64 = character code for the @ key; 251 = character code for √)

ASSIGN Command

Type: External

Purpose: To cause requests for a specific drive to be transferred to a different drive.

Syntax: ASSIGN [[*x* = *y*]...]

Comments: This command is used to route all requests for a specified drive to a different drive. It allows you to use program applications that have preassigned drives (such as A: and B:) with any other drive (such as C:).

If the ASSIGN command is entered without parameters, the system uses the normal drive assignments.

NOTE: The ASSIGN command will be ignored if a DISKCOPY or DISKCOMP command is requested.

Example:

```
ASSIGN A = C
```

The system routes all commands for Drive A: to Drive C:. This means that after the reassignment, if you enter DIR A:, for example, the system displays a list of all the files in Drive C:.

BACKUP Command

Type: External

Purpose: To back up files from a fixed disk to a diskette.

Syntax: BACKUP [d:] [*pathname*] [*filespec*] *d*:[/S][/M][/A] [/D:*mm-dd-yy*]

NOTE: You **must** include at least one of the following three parameters: *d*:, *pathname*, or *filespec*.

Comments: When the BACKUP command is executed, the back-up operation begins by displaying the following message:

```
Insert diskette 01 in drive d:
Warning! Diskette files will be erased
Strike any key when ready
```

The back-up operation backs up onto the diskette all files matching the entered file specification, displaying the name of each file as it is backed up. As each diskette is filled with data, BACKUP prompts you to insert the next diskette in the series. The back-up operation is complete when all specified files have been backed up.

After the appropriate diskette is inserted, press any key to continue.

When backing up files to a series of diskettes, it is important to label each diskette with a sequence number so that the proper order of the diskettes can be maintained in subsequent RESTORE operations.

When adding files to a series of diskettes that contain previously backed-up files, the additional files can only be added to the last diskette in the series. If a diskette other than the last in the series is inserted, the following error message appears:

```
Insufficient disk space
```

and the back-up operation terminates.

Switches for BACKUP are:

/S Backs up all matching files in the specified directory and all the following subdirectories. If /S is not specified, only the files in the specified directory are backed up.

/M Backs up all matching files that have been modified since the previous back-up.

/A Adds the back-up files to the diskette that is in the specified drive. The diskette inserted **must** be a back-up diskette used in a previous back-up operation. If the diskette is not a back-up diskette, the following message appears:

```
Diskette is not a backup diskette
```

and the back-up operation terminates.

NOTE: If you do not use /A, all files on your back-up diskette are erased before the new back-up files are written.

/D Backs up all matching files that have been modified on or after a specified date. See the DATE command for the date format.

NOTE: Files placed on a diskette with the BACKUP command can only be accessed with the RESTORE command. Use the COPY command to create files that you want to access with commands other than RESTORE.

To back up files from a fixed disk to a diskette, use a formatted diskette.

If no pathname is specified, the system backs up all the files that are in the working directory.

The wild card characters ? and ∗ can be used in the file specification.

BACKUP sets EXIT codes as follows:

0 = Normal completion
1 = No files found
3 = Terminated by ∧BREAK
4 = Terminated because of an error

Examples:

```
BACKUP C:\SUBDIR\USERS\SOFTWARE\TEXT. TXT   B:/M
```

This command backs up the file TEXT.TXT in the specified subdirectory from the disk in Drive C: to the diskette already in Drive B:, assuming the TEXT.TXT file was modified since the last back-up.

```
BACKUP  C:   A:/S
```

This command backs up all files, starting with the working directory and all subdirectories in that path, to the diskette in Drive A:.

```
BACKUP \NOTES A:
```

This command backs up all files in the NOTES directory of the current fixed disk drive to the diskette in Drive A:.

```
BACKUP C:\NOTES\∗.TXT A:/S
```

This command backs up all files with the extension .TXT in the NOTES directory, and all subdirectories in that path to the diskette in Drive A:.

BREAK Command

Type: Internal (Configuration)

Purpose: Turns extended ^BREAK checking on or off.

Syntax:

```
BREAK  [ON|OFF]
or

BREAK=[ON|OFF]
```

Comments: When you enter BREAK without parameters, the screen displays whether BREAK is currently on or off.

The BREAK command enables or disables extended ^BREAK checking. Normally, MS-DOS checks for the ^BREAK keys only during screen, keyboard, or printer I/O. By executing a BREAK ON command, you can cause MS-DOS to check for ^BREAK during every operation (including disk access). This is useful when you need the ability to stop a non-screen-intensive program (such as an assembler or a compiler) during its execution.

You can also enable extended ^BREAK checking by using the BREAK ON command in your configuration file (see Section 4-7, HOW TO CONFIGURE YOUR SYSTEM).

BUFFERS Command

Type:	Internal (Configuration)
Purpose:	To set the number of disk buffers allocated to memory.
Syntax:	BUFFERS =xx

Comments: Buffers set aside space in the computer's memory. The amount of space in the buffers determines the amount of holding buffers used during read/write requests. When read/write operations are requested, a check is performed to see if that record is already in the buffer. If so, then a disk access is saved. The buffers are allocated by a first in, last used method.

xx must be a value from 1 to 99. The default is two disk buffers.

A minimum of three disk buffers is recommended for fixed disk applications, and 10 to 12 buffers are recommended for data base applications. If you set the buffers for more than 20, it will cause the check for records to take longer and can, therefore, slow down operations.

Each additional buffer specified will increase the apparent size of MS-DOS in memory by 528 bytes.

To set buffers, include a BUFFERS =xx command in your CONFIG.SYS file (see Section 4-7, HOW TO CONFIGURE YOUR SYSTEM).

CHDIR (CHANGE DIRECTORY) Command

Type: Internal

Purpose: To display the working directory or to change the working directory.

Syntax:

```
CHDIR   [[d:]pathname]
or
CD   [[d:]pathname]
```

Comments: If no parameters are entered, the screen displays the working directory.

To change from the working directory to another directory, enter the command and the pathname.

Example:

If the working directory is \SUBDIR\USER\SERVICE and you want to change to another directory, such as \SUBDIR\USER\SERVICE\FORMS, entering:

```
CD \SUBDIR\USER\SERVICE\FORMS
```

puts you in the new directory. Using the shorthand notation .. always puts you in the parent directory of the working directory.

CHKDSK (CHECK DISK) Command

Type: External

Purpose: To scan the directories on a diskette or in a partition, and check them for consistency.

Syntax: CHKDSK [d:][*filespec*][/F][/V]

Comments: Switches for CHKDSK are:

/F	Corrects some errors found in the directory. Other errors are reported, but not corrected. These errors are corrected only when you answer Y to the correction request message. If /F is not specified, CHKDSK prepares to correct the disk and reports the errors, but does not make the corrections.
/V	Displays messages indicating how CHKDSK is progressing, and also displays more detailed error messages. For example, the file specification (including pathname) is displayed as the directory entries are being checked. Hidden files are also listed by this option.
filespec	Displays the number of non-contiguous areas that this file(s) occupies. Only files in the working directory are checked with CHKDSK.

Run CHKDSK occasionally on each diskette or the MS-DOS partition to verify that the directory structure is correct. If any errors are found, the appropriate error message is displayed.

After the diskette or partition has been checked, a status report appears. For example, the screen might display:

```
322560 bytes total disk space
  9216 bytes in 2 hidden files
311296 bytes in 8 user files
  2048 bytes available on disk

131072 bytes total memory
118688 bytes free
```

CHKDSK assumes that the diskette or partition being checked is already installed. On single-drive systems, you must specify a drive different from the default drive (unless you are checking the MS-DOS diskette itself).

To redirect the output from CHKDSK to a file, enter:

```
CHKDSK  A:>filespec
```

NOTE:

- If invalid clusters (allocation units) are found, the clusters are listed, and the message "CONVERT LOST CHAINS TO FILES?" is displayed. If you answer Y, and had specified /F in the command, each lost allocation unit chain is recovered into a file in the root directory. The naming convention used for these files is FILEXXXX.CHX, where XXXX is a sequential number (starting at 0000). These files can be examined to determine if they contain any useful data. If they do not, they can be erased.

- System performance can be impaired by badly fragmented files, as the files cannot be sequentially read by MS-DOS. Use CHKDSK *.* to determine the degree of fragmentation of the files on the disk.

- Care should be used when redirecting output because all error messages and user prompts are also redirected to the output file. Because of this, it is better if you do not use the /F option when redirecting output.

CLS Command

Type: Internal

Purpose: To clear the screen.

Syntax: CLS

Comments: When executed, CLS clears the screen.

COMP (COMPARE) Command

Type: External

Purpose: To compare the contents of one file to that of another file, or sets of files to sets of files.

Syntax: COMP [*d:*][*pathname*][*filespec*]
[*d:*][*pathname*][*filespec*]

NOTE: For the first entry, you **must** include at least one of the following parameters: *d:*, *pathname,* or *filespec.*

Comments: Use COMP to compare individual files or sets of files. You may use wild card characters in the file specification, or specify only the drive or pathname, to compare sets of files. Use DISKCOMP to compare entire diskettes on a track-by-track basis.

The files being compared may reside on the same or different diskettes. The entries are optional. If they are omitted, MS-DOS prompts you for them.

Since COMP compares the files byte-for-byte, any bytes that do not match are displayed in an error message. The error message displays the hexadecimal location (offset) and the value of the bytes that did not compare.

An error message might look like this:

```
Compare error at offset 12A

File 1 = 32

File 2 = 20
```

After 10 unequal comparisons, COMP aborts the comparison of that file, and the following message appears:

```
10 Mismatches - ending compare
```

When all requested file comparisons have been completed, the screen displays:

```
Files compare ok

Compare more files    (Y/N)?
```

This permits you to do additional comparisons.

When comparing files where one or both files do not contain an end-of-file mark, the message "EOF mark not found" is also displayed.

NOTE: The computer does not make a comparison if the file sizes are different.

COPY Command

Type: Internal

Purpose: To copy a file to the same or another disk, or to transfer data between devices.

Syntax: COPY [/A][/B][d:][*pathname*][*source filespec*]
[/A][/B][d:][*pathname*][*destination filespec*]
[/A][/B][/V]

or

COPY [/A][/B][d:][*pathname*][*source filespec*]
[/A][/B][+ [d:][*pathname*]*source filespec*
[/A][/B]...][d:][*pathname*][*destination filespec*]
[/A][/B][/V]

NOTE: You **must** include either the *pathname* or *filespec* parameters.

Comments: When copying to a different directory or a different disk, the copied file may be given a different name. Files being copied into the same directory **must** be given different names.

Concatenation (combining files) may be performed during the COPY process.

Switches for COPY are:

/V	Verifies the copy.
/A or /B	Indicates the amount of data to be processed. These switches can be placed before or after any specified *filespec*.

/V Switch:

The /V switch causes MS-DOS to verify the copy. The COPY command runs slower when you select this option so that the system can verify each file copied onto the disk.

/A and /B Switches:

The switches /A and /B indicate the amount of data to be processed. Each applies to the file specification preceding it, and to all remaining file specifications on the command line until another /A or /B is encountered. These switches can have the following meanings:

When used with a source file specification:

/A causes the file to be treated as a text (ASCII) file. The file's data is copied up to, but does not include, the first end-of-file (EOF) character found in the file. The remainder of the file is not copied.

/B causes the entire file to be copied.

When used with a destination file specification:

/A causes an EOF marker (^Z) to be added as the last character of the file. /A is a default value when files are being concatenated (see Option 4 in this section).

/B causes no EOF marker to be added to the end of the file. /B is the default when concatenation is not being performed (see Options 1, 2, and 3).

Wild Card Characters

The wild card characters ? and * may be used in parameters for both the source and destination file specifications. When using a ? or * in the source file specification, the names of the files are displayed as they are copied.

Example:

```
COPY MYFCST.'82 B:*.JAN
```

copies the file MYFCST.'82 from the default drive (Drive A:) to the diskette in Drive B:, and renames the copy MYFCST.JAN.

The COPY command has four format options. A description of each one follows:

Option 1: Copying a File With the Same Name as the Source File Onto a Different Directory or Disk

In this option, the file is copied onto another directory or disk in the specified drive. The name of the target file is the same as the source file because a new name was not specified. You cannot copy a file with the same name as another file onto the same disk without specifying a different directory.

Examples:

```
COPY  SOURCE  \SUBDIR
```

copies the file SOURCE that is on the disk in the default drive into the directory SUBDIR on the same disk, and gives the duplicate file the same name.

```
COPY SOURCE B:
```

copies the file SOURCE that is on the disk in the default drive onto the disk in Drive B:, and gives the duplicate file the same name.

Option 2: Copying a File With a Different Name From the Source File Onto the Same Disk

This option is used when the copied file is to have a different name from the source file, but they are to be on the same disk.

Example:

```
COPY MINE YOURS
```

The file named MINE is copied onto the disk in the default drive, but is renamed YOURS.

Option 3: Copying a File With a Different Name From the Source File Onto a Different Disk

This option is used when the copied file is to have a different name from the source file and is to be copied onto another disk in the specified drive.

Example:

```
COPY SOURCE B:COPIED
```

In this example, the file named SOURCE is copied onto the disk in Drive B:, and is renamed COPIED.

Reserved device names may be used for the copy operation.

Examples:

```
COPY CON: AUX
COPY FILEA CON:
COPY AUX: LPT1
```

NUL: can also be used as shown above.

Option 4: Combining Two or More Files on the Same or Different Disks, and Assigning the New File a Name (Concatenation)

You must combine two or more files into one file by adding the additional files to the end of the first. The date and time recorded for the resulting file are the date and time that you combined the files.

To combine files, list the source files and separate them by plus signs.

Example:

```
COPY AFILE.XYZ+BFILE+B:CFILE.TXT
  ONEBIG.FIL
```

In this example, the COPY command now creates the new file, ONEBIG.FIL, on the disk in the default drive. AFILE.XYZ, BFILE and CFILE.TXT. (on Drive B:) have now been combined and deposited into the file ONEBIG.FIL.

If a destination file isn't specified, the additional files are added to the end of the first file.

Concatenation is normally performed in text (ASCII) format. To combine binary files, use the /B switch. This forces the COPY utility to use the physical EOF mark. You may combine ASCII and binary files through the use of either the /A (ASCII format) or /B (binary format) switches.

Examples:

```
COPY A.EXT + B.EXT/B + B:C.EXT/ A
  BIGFILE
```

An /A or /B switch affects the file it is placed after, and affects all subsequent files on the command line, until another /A or /B is found.

In this example, B.EXT is copied in its entirety, along with A.EXT and C.EXT (which is treated as a text file) on the disk in Drive B: into BIGFILE on the disk in the default drive.

You may use the wild card characters ? and ∗ in the file specifications of both the source and destination files.

```
COPY ∗.LST RESULT.AAA
```

The result of this example is that all files matching ∗.LST are combined into one file called RESULT.AAA.

To enlarge on this concept, the following example combines all files with the extension of .MAN and .$$$ into one file called LABOR.FST.

```
COPY ∗.MAN + ∗.$$$ LABOR.FST
```

If you attempt to copy a file into the same directory without specifying a new name, MS-DOS detects the error and aborts the COPY command. When you are concatenating, as each source file is found, its name is compared to the name of the target file. If the name is the same, the file is skipped and the message "Content of destination lost before copy" is displayed. Further concatenation proceeds normally. To avoid losing the existing data in the file to which you want to concatenate data, enter the destination file specification first in the command line, as in the following example:

```
COPY REF.LST + *.LST
```

The result is that all *.LST files, except REF.LST, are appended to the file REF.LST.

Special Considerations of Concatenation

Example:

```
COPY   B:FORECAST.'83 + ,,
```

In this example, the system copies the file FORECAST.'83, located on the diskette in Drive B:, to the default disk, giving it a new date and time.

When you are concatenating, the computer considers the copying process to be successful if at least one, but not necessarily all, of the named source. files are found. If no source files are found, you receive the message, "0 File(s) copied".

Changing the Date of a File

To change the date and time recorded for a file to the current date and time, use a command like the following:

```
COPY B:MYFILE.AAA+,, B:
```

NOTE: The two commas are necessary to define the end of the source file specification because the computer expects to see another file specification after the plus sign.

CTTY Command

Type:	Internal
Purpose:	To change the standard input and output device to an auxilliary device, or to return to the original device.
Syntax:	CTTY *device*

Comments: *device* represents the device used to issue the commands to MS-DOS.

Example:

When you enter:

```
CTTY AUX
```

all standard input and output is moved to an asynchronous communications port. To return to the original device, enter:

```
CTTY CON
```

This moves the command input and output back to the computer keyboard and screen.

Acceptable device names are listed following:

Device:	**Meaning:**
AUX: or COM1:	Primary asynchronous communications port.
COM2:	Secondary asynchronous communications port.
CON:	Keyboard input and screen output.

DATE Command

Type: Internal

Purpose: To display and set the date.

Syntax: DATE [*mm-dd-yy*]

Comments: If you enter DATE without a parameter, a message similar to this is displayed:

```
Current date is Mon 12-01-1980.
Enter new date:
```

Press ENTER if you do not want to change the date shown.

Optionally, the date may be given as a parameter to the DATE command, as in:

```
DATE 3-9-81
```

In this case, no message appears.

The new date must be entered using numerals only. Letters are not permitted. The allowable parameters are:

mm = 1-12
dd = 1-31
yy = 80-99 (1980-2099)
or 1980-2099

The date, month, and year entries can be separated by hyphens (-) or slashes (/).

If the parameters or separators are illegal, MS-DOS returns the message:

```
Invalid date
Enter new date:
```

The computer then waits for you to enter a legal date.

DEL (DELETE) Command

Type: Internal

Purpose: To delete specified files from a disk.

Syntax: DEL [*d*:][*pathname*][*filespec*]

NOTE: You **must** include at least one of the following parameters: *pathname* or *filespec*.

Comments: If the file specification given for DEL is the wild card character name *.*, the prompt "Are you sure (Y/N)?" appears. If you type Y in response, all files in the working directory are deleted.

The ERASE command works the same as the DEL command.

DEVICE Command

Type: Internal (Configuration)

Purpose: To direct MS-DOS to load an additional file that contains a device driver.

Syntax: DEVICE [*d*:][*pathname*]*filespec*

Comments: For example, if you wanted to use extended screen and keyboard controls, the command:

```
DEVICE = ANSI.SYS
```

must be placed in your CONFIG.SYS file (see Section 4-7, HOW TO CONFIGURE YOUR SYSTEM).

The data file to be loaded with the DEVICE configuration command is an assembly language program called a device driver.

DIR (DIRECTORY) Command

Type:	Internal
Purpose:	To list the files in a directory.
Syntax:	DIR [*d*:][*pathname*][*filespec*][/P][/W]
Comments:	Switches for DIR are:

/P	Causes the screen display to pause when the screen is full. Press any key to continue the directory listing.
/W	Produces a wide screen display of the directory that lists file specifications only. Each line contains five file specifications. Use this switch only on an 80-column display.

If no parameters are entered, all directory entries in the working directory on the disk in the default drive are listed.

If only the drive specification is entered (DIR *d*:), all entries in the working directory on the disk in the specified drive are listed.

If only a filename is given, with no extension, all files with that filename on the disk in the default drive are listed.

If a full file specification is given, only the file with that specified filename and extension is listed.

Wild Card Characters

The wild card characters ? and * may be used in the file specification. If either the filename or its extension is omitted, the wild card character * is assumed. The following DIR commands are equivalent:

```
DIR          DIR  *.*
DIR FILE     DIR FILE.*
DIR .EXT     DIR *.EXT
```

NOTE: DIR .EXT can only be used for the working directory. The other wild card examples can be used with a pathname.

Listing All Files

Use this option to list all the files in the working directory.

```
DIR
or
DIR d:
```

The result will look similar to the following:

```
A>DIR

FORECAST QTR     100    11-17-83     9:54p
FCST      A     1099    12-01-82     3:00a
LABOR          20123     2-28-83    12:13p
```

Use this option to list all files in a subdirectory:

```
DIR pathname
or
DIR d:pathname
```

The first example (DIR *pathname*) lists all files in the specified subdirectory in the default drive. The second example (DIR *d:pathname*) lists all files in the specified subdirectory of a specified drive.

Listing Selected Files

Use this option to list selected files in a directory:

DIR [*pathname*]*filespec*
or
DIR *d*: [*pathname*]*filespec*

Examples:

```
DIR  .TXT
```

displays a directory listing of all files in the working directory with an extension of .TXT.

```
DIR  \SUBDIR\ACCOUNTS
```

displays the directory listing of all files with the filename ACCOUNTS, regardless of extension, in the SUBDIR subdirectory.

```
DIR  \SUBDIR\*. '83
```

displays the directory listing of all files in the SUBDIR subdirectory with an extension of . '83.

DISKCOMP (DISKETTE COMPARE) Command

Type: External

Purpose: To compare the contents of one diskette to the contents of another diskette.

Syntax: DISKCOMP [*d*: [*d*:]][/1][/8]

Comments: Switches for DISKCOMP are:

/1 Compares only the first sides of the diskettes, even if the diskettes and drives are double-sided.

/8 Compares eight sectors per track, rather than nine.

You may specify the same or different drives. If the drives are the same, if you omit both of the drive designators, or if you specify only one drive and it is the default drive, a single-drive comparison is performed. You are prompted to insert the correct diskette at the appropriate time and press any key to continue the comparison.

If a drive designator is omitted from the command line, the default drive is assumed. If your default drive is the fixed disk, and you enter DISKCOMP without a drive designator, an error message is issued because DISKCOMP is invalid for the fixed disk.

When you enter:

```
DISKCOMP A: B:
```

you receive the prompt:

```
Insert first diskette in drive A:

Insert second diskette in drive B:

Strike any key when ready
```

If you have already inserted the diskettes in the drives, press any key to begin the comparison. As much data as can be stored in memory is read from the first diskette and compared to the second diskette. Then the system reads another block of data from the second diskette and compares it to the first diskette. This process continues until all the data on both diskettes has been compared. Then the screen displays:

```
Diskettes compare ok

Compare more diskettes    (Y/N)?
```

Press Y to compare another pair of diskettes, or N to end the comparison.

DISKCOMP compares all 40 tracks on a track-for-track basis, and issues a message if the tracks are not the same. The message indicates the track number and the side where a mismatch was found.

When a diskette containing files created by the COPY command is compared to the diskette the files were copied from, DISKCOMP may not issue a "Diskettes compare ok" message. COPY produces a copy that contains the same information, but places the information in different locations on the target diskette from those on the source diskette. Use the COMP command to compare individual files on different diskettes.

If a disk error occurs while DISKCOMP is running, a message indicates where (track and side) the error occurred. Then DISKCOMP continues to compare the rest of the diskette.

DISKCOPY (DISKETTE COPY) Command

Type: External

Purpose: To copy the contents of one diskette to another.

Syntax: DISKCOPY [*d*: [*d*:]][/1]

Comments: The switch for DISKCOPY is:

/1 Copies only the first side of the diskette, even if the diskette or drive is double-sided.

You may specify the same or different drives. If the drives are the same, if you omit both of the drive designators, or if you specify only one drive and it is the default drive, a single-drive copy operation is performed. You are prompted to insert the correct diskette at the appropriate time and press any key to continue the copy.

If a drive designator is omitted from the command line, the default drive is assumed. If your default drive is the fixed disk, and you entered DISKCOPY without a drive designator, an error message is issued because DISKCOPY is invalid for the fixed disk.

When you enter:

```
DISKCOPY A: B:
```

the screen displays:

```
Insert source diskette in drive A:
Insert target diskette in drive B:
Strike any key when ready
```

If you have already inserted the diskettes in the appropriate drives, press any key to begin the copy procedure. As much data as can be stored in memory is taken from the source diskette and copied to the target diskette. Then the system goes back to the first diskette for more data. This process continues until all the data on the source diskette has been copied to the target diskette. Then the screen displays:

```
Copy complete

Copy another   (Y/N)?
```

Press Y to copy another diskette, or N to end the copy procedure.

DISKCOPY automatically determines the number of sides to copy, based on the source diskette. If only the first side of the source diskette can be read, then only the first side is copied. If the source diskette is double-sided, both sides are copied, unless you specify the /1 switch. If any incompatibilities exist, the message "Incompatible drive types" appears.

Diskette Fragmentation Problems

Diskettes having a large amount of file creations and deletions tend to become fragmented. This fragmentation occurs because diskettes are not designed to allocate space sequentially. The first free sector found is the next sector allocated, regardless of its location on the diskette.

Fragmentation often causes degraded performance because of the excessive head movement involved in finding, reading, and/or writing a file.

In this case, it is much better to use COPY, instead of DISKCOPY, to eliminate the fragmentation.

You can easily copy all files on a diskette by using the following example:

```
COPY A:*.* B:
```

This command copies all files from the diskette in Drive A: to the diskette in Drive B:, and assigns them the same names.

ECHO Command

Type:	Internal (Batch)
Purpose:	To turn the batch echo feature on and off.
Syntax:	ECHO [ON|OFF|*message*]
Comments:	Usually when COMMAND.COM encounters a batch file command, the screen displays it. ECHO, plus the ON and OFF parameters, turns the screen display on or off, allowing you to control the display. If ECHO is entered without parameters, the screen displays the current setting (either ON or OFF). If *message* is used, the message is displayed regardless of the current setting.

ERASE Command

Type: Internal

Purpose: To delete a specified file from a disk.

Syntax: ERASE [*d*:][*pathname*][*filespec*]

NOTE: You **must** include at least one of the following parameters: *pathname* or *filespec*.

Comments: The DEL command can be substituted for the ERASE command.

The wild card characters ? and * can be used in file specifications. Use wild card characters carefully. Multiple files can be erased with a single command.

To erase or delete all files in the working directory, enter:

```
ERASE [d:]*.*
```

When you specify the file specification *.* to erase all files in the working directory, MS-DOS issues the prompt, "Are you sure (Y/N)?" If you enter Y, all the files in the working directory on the disk in the specified drive are deleted.

EXE2BIN Command

Type: External

Purpose: To convert files from the .EXE format to the .COM format.

Syntax: EXE2BIN [*d:*][*pathname*]*filespec* [*d:*][*pathname*][*filespec*]

Comments: The first parameter is the input (.EXE) file. If no extension is given, it defaults to .EXE.

The second parameter is the output (.COM) file. If no specific drive is given, the drive of the input file is used. If no file specification is given, the file specification of the input file is used. If no extension is given, .BIN is used.

The input file must be in valid .EXE format, produced by the MS-LINK utility. The resident, or actual, code and data contained in the file must be less than 64 Kbytes. Also, there must be no stack segment.

Two kinds of conversions are possible, depending on whether you specify CS:IP.

If CS:IP is not specified, a pure binary conversion is assumed. If segment fix-ups are necessary, the following prompt appears:

```
Fix-up needed - base segment (hex):
```

If you enter a legal hexadecimal number, execution continues.

If CS:IP is specified as 100H, it is assumed the file is to be run as a .COM file ORGed at 100H, with the first 100H of the file deleted. No segment fix-ups are allowed, because .COM files must be segment relocatable.

If CS:IP does not meet one of these criteria, or meets the .COM file criterion but has segment fix-ups, the following error message is displayed:

```
File cannot be converted
```

To produce standard .COM files, you must both ORG the file at 100H and specify the first location as the start address. This is done using the END statement. For example:

```
ORG          100H
START:
.
.
.
END        START
```

FDISK Utility

Type: External

Purpose: To install a separate data area or partition for MS-DOS on a fixed disk.

Syntax: FDISK

Comments: To install a partition on a fixed disk to house the MS-DOS operating system, files, and subdirectories, enter:

```
FDISK
```

The screen displays the FDISK menu shown following:

```
COMPAQ Personal Computer
Fixed Disk Management Version 1.XX
(C)Copyright COMPAQ Computer Corp. 1983

* FDISK Menu *

Current Fixed Disk Drive:      1
Available Fixed Disk Drives:   1

Choose one of these options:

   1.  Install MS-DOS Partition
   2.  Modify Active Partition
   3.  Remove MS-DOS Partition
   4.  List Partition Information

Enter your choice: [1]

Press ESC to return to MS-DOS
```

For more information on FDISK, see Chapter 2, FDISK UTILITY.

FILES Command

Type: Internal (Configuration)

Purpose: To specify the number of files that can be simultaneously open at any one time.

Syntax: FILES = xx

Comments: xx must be a value from 1 to 99. The default value is 8.

Use FILES to increase the number of concurrently-open files, particularly when applications programs return errors indicating insufficient files.

This is accomplished by including a FILES = xx command in your CONFIG.SYS file (see Section 4-7, HOW TO CONFIGURE YOUR SYSTEM).

FIND Command

Type: External (Filter)

Purpose: To search for a string of text in a file.

Syntax:

```
FIND [/V][/C][/N]"string"
  [[d:][pathname]filespec...]
```

Comments: Switches for FIND are:

/V Displays on the screen all lines that don't contain the specified string of text.

/C Displays the number of lines that contain a match in each of the files.

/N Displays the line number followed by each line in the file that contains a match.

FIND is a filter that displays the lines containing a specific string of text in specified files.

If no parameters are entered, FIND displays all lines in the specified file that contain the specified string.

When you type the string, enclose it in quotation marks (").

If the string itself includes quotation marks, use the following syntax:

```
FIND ""string""
```

Examples:

If you want to locate all occurrences of “He said” in a file, enter:

```
FIND   “ “ “He said” ” ” TEXTFILE
```

When you enter:

```
FIND   “Fool’s Paradise” BOOK1.TXT
BOOK2.TXT
```

the screen displays all lines from BOOK1.TXT and BOOK2.TXT respectively that contain the string, Fool’s Paradise.

When you enter the command:

```
DIR   B:|FIND/V“DAT”
```

the screen displays all the names of the files on the diskette in Drive B: which do not contain the string, DAT.

FOR Command

Type: Internal (Batch)

Purpose: To repeat a command for several variables.

Syntax: FOR %%*c* IN (*set*) DO *command*

Comments: c can be any character except 0 through 9.

The FOR command executes repeatedly for every variable listed in *set*.

If the wild card characters * and ? are used in *set*, the variable is set to each matching pattern from the disk. Therefore, there can be only one item in a set. If more than one is entered, the other items are ignored.

When batch processing, use double percent signs (%%) so that one % is left. If you enter only one % followed by a variable, the batch processor reads the % and the variable and deletes it as an error. The FOR command, therefore, would never see it.

Examples:

```
FOR %%f IN(FILE.1 FILE.2) DO DIR %%f
```

This command causes the system to display a directory listing for FILE.1 and FILE.2. The f variable is replaced with the first item in the set (FILE.1), the directory listing is displayed for that file, and then the second item (FILE.2) is used.

```
FOR %%g IN(I*.TXT) DO REN%%g *.DOC
```

This command causes the system to rename all files that begin with I, have any next characters, and have an extension of .TXT to the same filename, but with an extension of .DOC. The g variable is replaced with a file specification that matches the item in the set, the file is renamed, and then the next match is sought.

FORMAT Command

Type: External

Purpose: To set up a diskette or partition for use with MS-DOS. FORMAT also analyzes the entire disk for any defective tracks, and prepares the disk to accept MS-DOS files. It initializes the directory and sets up the file allocation table and the system loader. It is also used to assign a volume label.

Syntax: FORMAT [*d*:][/S][/1][/8][/V][/B]

Comments: Switches for FORMAT are:

/S Copies the system files from the source disk to the new destination disk in the following order:

IBMBIO.COM
IBMDOS.COM
COMMAND.COM

/1 Formats the destination diskette for single-sided use.

/8 Formats the disk with eight sectors per track. You can put the system files on the diskette with the SYS command, but you can only put version 2 of MS-DOS on a diskette formatted with the /8 switch.

/V Displays the volume label prompt and allows you to name the volume for your diskette or partition.

/B Formats the disk with eight sectors per track, and leaves space for the operating system (hidden) files. This allows you to put any version of MS-DOS on the diskette.

NOTE: The /B switch cannot be used with /V or /S.

The MS-DOS partition on the fixed disk must be formatted with the FORMAT command. Diskettes must be formatted with either FORMAT or DISKCOPY before they can be used with MS-DOS.

If the drive specifier is omitted from the command line, the default drive is assumed.

NOTES:

- IBMBIO.COM and IBMDOS.COM are hidden from operator access. As such, they do not appear in any directory searches, including the DIR command.
- Formatting destroys all existing data on a diskette or in a partition.
- Defective tracks are marked as reserved during the formatting process to prevent bad tracks from being allocated to a data file.
- FORMAT produces a status report, indicating the following:

  Total disk space.
  Space marked as defective.
  Space currently allocated to files (for use with /S).
  Available space.

- FORMAT can be used to format double-sided diskettes for single- or double-sided use. Single-sided diskettes can be formatted with the /1 option. Double-sided diskettes formatted for single-sided use can be used with a single- or double-sided drive.

Example:

The following command causes the diskette in Drive B: to be formatted, and the system files to be copied:

```
A>FORMAT  B:/S
```

The screen displays:

```
Insert new diskette for drive B:

and strike any key when ready
```

After you insert the appropriate diskette and press any key, the screen displays:

```
Formatting...
```

Once the formatting process is complete, the screen displays something similar to this:

```
Formatting...Format complete

System transferred


362496  bytes total disk space
 48960  bytes used by system
321536  bytes available on disk

Format another  (Y/N)?
```

If you included the /V option, before the disk statistics are displayed, the screen displays the following message:

```
Volume label (11 characters, ENTER for none)?
```

You may enter up to 11 characters for the volume label, or press ENTER if you do not want to assign a volume label.

NOTE: You must assign the volume label at this point. A label cannot be added later.

GOTO Command

Type:	Internal (Batch)
Purpose:	To direct the system to continue processing at the command that appears after the label.
Syntax:	GOTO *label*
Comments:	GOTO causes processing to continue, starting with the line after the *label* parameter. If a label isn't entered, the current batch file is terminated.

Batch processing ignores lines in a batch file that start with a colon (:). For example, the following batch file:

```
:foo
REM looping...
GOTO foo
```

causes the message "REM looping..." and "GOTO foo" to be displayed repeatedly.

GRAPHICS Command

Type: External

Purpose: To print a graphics display on a compatible graphics printer.

NOTE: See your Authorized COMPAQ Computer Dealer for a list of compatible graphics printers.

Syntax: GRAPHICS

Comments: To print a graphics display, such as a bar graph, enter:

```
GRAPHICS
```

Now press SHIFT + PRTSC. The printing may take as long as three minutes to complete.

This command increases the size of MS-DOS in memory by 736 bytes. The command only needs to be executed once, after you turn your computer on. Therefore, SHIFT + PRTSC will cause the graphics screen to be printed.

NOTE: In high resolution, the system prints the screen sideways on the page. That is, the upper-right corner of the screen is in the upper-left corner of the printed page. In medium resolution, the screen is printed in as many as four shades of gray.

If the screen is in one of the alphanumeric modes, the screen is printed via the default print screen function, as if the GRAPHICS command had not been executed.

IF Command

Type: Internal (Batch)

Purpose: To execute commands that depend on a certain condition.

Syntax: IF [NOT] *condition command*

Comments: The system executes *command* when *condition* is true, and ignores *command* when *condition* is false.

condition may be:

- ERRORLEVEL *number*, when *number* is one of the following:

  0 = Normal completion
  1 = No files found
  3 = Terminated by ^BREAK or ESC
  4 = Terminated because of an error

  ERRORLEVEL *number* is true if the last program or command that was executed had an exit code greater than or equal to *number*.

- *string1* = = *string2* is true if *string1* matches *string2*.

- EXIST *filespec* is true if the specified file exists.

If you enter the NOT parameter, *command* is executed when *condition* is false.

BACKUP and RESTORE are the only commands that set the exit code.

MKDIR (MAKE DIRECTORY) Command

Type: Internal

Purpose: To create a new subdirectory.

Syntax: MKDIR [*d:*]*pathname*

or

MD [*d:*]*pathname*

Comments: This command allows you to create subdirectories in a specified path. You can create subdirectories by entering MKDIR, a drive specifier, a pathname, and the name of the subdirectory you are creating.

Examples:

When you enter:

```
MKDIR     \USER
```

the subdirectory USER, in the root directory of the disk in the default drive, is created. To create a subdirectory in the current directory, enter the command and the name of the directory you are creating.

When you enter:

```
MD  SERVICE
```

the subdirectory SERVICE, under the working directory, is created.

When you enter:

```
MKDIR  B:\USER\SERVICE\MEMOS
```

the subdirectory MEMOS, under the subdirectory SERVICE, under the subdirectory USER, in the root directory on the disk in Drive B:, is created.

MODE Command

Type: External

Purpose: To set the operations mode for the display screen, the asynchronous communications port, and the parallel printer, and to redirect printer output to an asynchronous communications port. It is also used to set the keyclick duration and to enable/disable an external monitor.

NOTE: When the MODE command is used to set up printer redirection and asynchronous communications port parameters, intercept code is made resident in memory, which increases the size of MS-DOS in memory by 432 bytes.

Syntax: MODE LPT#:[*n*][,*m*][,P]

or

MODE *n* [,*m*[,T]]

or

MODE COM*n*:*baud*[,[*parity*],[*databits*],[*stopbits*]
[,P]]

or

MODE LPT#: = COM*n*:

or

MODE CLI[ck] [=] #

Comments: If there is an invalid or missing *n* or *m* parameter, the operations mode for that parameter does not change.

The MODE command has five possible formats, as follows:

Format 1, Printer:

```
MODE   LPT#:[n][, [m][,P]]
```

# = printer number (1, 2, or 3)
n = characters per line (80 or 132)
m = vertical spacing, or lines per inch (6 or 8)
P = retrying of time-out errors

If you specify the P option, time-out errors will be continuously retried until you terminate the retry loop by pressing ^BREAK. You can stop the time-out errors from being retried continuously by re-executing the MODE command without the P option.

Example:

```
MODE   LPT1:132,8
```

The normal defaults for a printer are 80 characters per line, and 6 lines per vertical inch. This example changes the operations mode for printer 1 to 132 characters per line, and 8 lines per vertical inch.

NOTE: The MODE command can be used to set these parameters only on compatible printers. See your Authorized COMPAQ Computer Dealer for a list of compatible printers.

Format 2, Display Modes and External Monitors:

MODE [*n*] [,*m*[,T]]

or

MODE [*n*] [,x]

n = characters per line (40, 80, BW40, BW80, CO40, CO80, or MONO).

40 – Sets the display width to 40 characters per line. The current state of color on an external composite monitor (the color burst signal) is unaffected.

80 – Sets the display width to 80 characters per line. The current state of color on an external composite monitor (the color burst signal) is unaffected.

BW40 – Sets the display width to 40 characters per line. Disables color on an external composite monitor.

BW80 – Sets the display width to 80 characters per line. Disables color on an external composite monitor.

CO40 – Sets the display width to 40 characters per line. Enables color on an external composite monitor.

CO80 – Sets the display width to 80 characters per line. Enables color on an external composite monitor.

MONO – Switches the active display to a compatible monochrome display adapter, if you have one installed. MONO causes an "Invalid parameter" error on a standard COMPAQ computer.

m = shift display right or left (R or L).

T = request for a test pattern (used to align display).

x = External or internal (E or I)

These parameters are used to enable and disable an external monitor (if one is connected). Including the E option causes the internal display to switch to a lower resolution character set (one which is compatible with the scan-rates of RGB and television monitors). Including the I option causes the internal display to switch to the (default) high resolution character set (which is not compatible with the scan-rates of RGB and television monitors), and usually disables any external monitor that is connected. These parameters are meaningful only in the 80-character display widths. The E option performs the same function as pressing ∧ALT+<(less than); the I option performs the same function as pressing ∧ALT+> (greater than).

This command is used to set the display width, and to enable or disable an external monitor. It is also used to shift the display one (for a 40-character line) or two (for a 80-character line) characters to the right or left to make viewing easier. When you include the T option in the MODE command, a prompt appears asking if the display is correctly aligned. If you respond with Y, the command ends. If you respond with N, the display shifts again, and the prompt is repeated.

NOTE: You cannot use both the *x* parameter and the *n*, *m* parameter combination; you must use one or the other.

Examples:

```
MODE  40,L,T
```

changes the operations mode to 40 characters per line, and shifts the display one character to the left. Then the test pattern is displayed, and the prompt appears. You can now shift the display again, without reentering the command, by responding to the prompt with N.

```
MODE  80,E
```

enables an external monitor to display 80 characters per line. The characters on the internal screen will look slightly different. This command could be included in an AUTOEXEC.BAT file, if you plan to always use an external monitor.

Format 3, Asynchronous Communications Port:

MODE COM*n*:*baud*[,[*parity*],[*databits*],[*stopbits*] [,P]]

n = Asynchronous communications port number (1 or 2).

baud = Baud rate (110, 150, 300, 600, 1200, 2400, 4800, or 9600). Enter only the first two numbers of the baud rate.

parity = None (N), odd (O), or even (E). The default is even (E).

databits = 7 or 8. The default is 7. When using 8 databits a none (N) parity must be used.

stopbits = 1 or 2. If *baud* = 110, the default is 2; otherwise, the default is 1.

These are called the protocol parameters, and are used to initialize the asynchronous communications port. You must always specify *baud* in the command. You can use the defaults for the other parameters by entering commas.

Examples:

```
MODE  COM2:30,N,8,2
```

sets the operations mode on asynchronous communications port 2 to 300 baud, odd parity, 8 databits, and 2 stopbits.

```
MODE  COM2:30,,,,P
```

In this command, the parity defaults to even, the databits default to 7, and the stopbits default to 1.

You can specify the P option only in command for a serial interface printer. The P option causes time-out errors to be retried continuously. To stop this loop, press ^BREAK. You can stop the time-out errors from being retried continuously by reinitializing the asynchronous communications port without the P option in the command.

Format 4, Printer Output to Asynchronous Communications Port:

```
MODE   LPT#:=COMn:
```

# = Printer number (1, 2, or 3).

n = Asynchronous communications port number (1 or 2).

Example:

```
MODE   LPT1:=COM2:
```

This command redirects all output for printer 1 to asynchronous communications port 2.

NOTE:

- You must initialize the asynchronous communications port (see Format 3) before using MODE to redirect parallel printer output to a serial device. You must also include the P option in the initialization command if the serial device is a printer.
- To stop the redirection of output for the printer designated with #, use:

  MODE   LPT#:[*n*][,*m*]

Format 5, Key Click Tone:

```
MODE   CLI[ck]   [=]   #
```

This command allows you to set the duration of the keyclick tone. The allowable range of values for # is from 0 (off/silent) to 127 (high/beep). The default value (when MS-DOS is initially loaded) is 1.

Example:

```
MODE   CLI=2
```

This command sets the duration of the keyclick tone to twice its default value, which causes the keyclick to sound louder.

MORE Command

Type: External (Filter)

Purpose: To display output one screen at a time.

Syntax: MORE

Comments: MORE is used to view a long data file one screen at a time. It prevents you from having to press ^NUM LOCK to stop the screen from scrolling. If there is more information to be displayed, —MORE— appears at the bottom of the screen. Press any key to display another screen of information. This process continues until all the input data has been read. ^BREAK can be used to terminate the request.

Example:

If you enter:

```
TYPE  MYFILES.COM¦MORE
```

or

```
MORE  <MYFILES.COM
```

the screen displays the MYFILES.COM file, one screen at a time.

PATH Command

Type: Internal

Purpose: To locate commands and batch files that are not in the working directory.

Syntax: PATH [[*d*:]*pathname*][;][[;[*d*:]*pathname*]...]

Comments: PATH tells MS-DOS which directories to search for a specific command or batch file after it searches the working directory.

Entering PATH without parameters causes the screen to display the current path. If you enter PATH and a semicolon (;), the system sets the search path to NUL, and searches only the working directory.

Specifying a nonexistent pathname does not result in an error. MS-DOS simply ignores the pathname, and proceeds to the next specified pathname.

Invalid parameters, such as an incorrect drive designator, are not detected until the path is searched.

Examples:

To search \BIN (working directory) and \BIN\USER\SERVICE (specified pathname) for a batch command, enter:

```
PATH \BIN\USER\SERVICE
```

To search several directories for an external command, enter several pathnames separated by semicolons. The search takes place in the order specified.

When you enter:

```
PATH  \BIN\USER\SERVICE;\BIN\USER\SALES;
\BIN\DEV
```

MS-DOS searches the working directory, and the subdirectories \BIN\USER\SERVICE, \BIN\USER\SALES, and \BIN\DEV for the external command.

PAUSE Command

Type: Internal (Batch)

Purpose: To suspend execution of a batch file.

Syntax: PAUSE [*comment*]

Comments: To allow time for changing diskettes or performing other actions during the execution of batch command files, enter the PAUSE command. PAUSE suspends execution until any key is pressed.

The optional *comment* may be entered on the same line as PAUSE to provide the operator of the batch file with some meaningful message during the pause. For example, you can enter a message to change the diskette in a designated drive. When PAUSE is encountered, the system suspends activity and displays the message:

```
Insert destination diskette in drive B:

Strike a key when ready...
```

When you press any key (except ^BREAK), execution of the batch file resumes. If you press ^BREAK, the system displays the prompt:

```
Terminate batch job   (Y/N)?
```

If you enter Y, execution of the remainder of the batch file is aborted, and control returns to the MS-DOS command level. If you enter N, execution of the batch file continues.

PAUSE can be used to divide a batch file into sections, ending the file at an intermediate point.

PRINT Command

Type: External

Purpose: To print a data file on a line printer while processing other MS-DOS commands.

Syntax: PRINT [[*d*:][*filespec*][/C][/P]...][/T]

Comments: Switches for PRINT are:

/C Turns the cancel mode on.

/P Turns the print mode on.

/T Deletes all files waiting to be printed, terminates the file being printed, and displays the message "All files canceled by operator".

The /C and /P switches apply to the preceding *filespec* and to all subsequent *filespecs* on the command line, until another switch is found.

If you enter the /C option for a file currently being printed, the printing of the file is cancelled, and the following message is output to the printer:

```
d:filespec  Canceled by operator
```

If the PRINT command is entered with file specifications, but with no options, the /P option is assumed.

The PRINT command increases the size of MS-DOS in memory by 3184 bytes.

PRINT is only used when there is a printer attached to your computer.

The first time you use the PRINT command, the screen displays the following message:

```
Name of list device [PRN:]
```

Specify a current device to become the output device. Pressing ENTER causes the PRN device (printer) to be used. Valid list devices include LPT1, LPT2, LPT3, COM1, COM2, COM3, AUX, PRN, etc.

If you are printing files from a diskette, make sure you do not remove the diskette from the drive before the printing has been completed.

Only files in the working directory can be printed. The current directory can be changed after the files have been added to the print queue.

More than one file can be entered with the PRINT command. The wild card characters ? and * may be used in the file specification.

A maximum of 10 files can be in the print queue at one time. If more than 10 files are requested for printing, the first 10 are entered in the print queue, and the remainder are ignored. The following error message appears:

```
PRINT queue is full
```

Files are printed in the order entered. A form feed is issued between each of the printed files. Tab characters are expanded with blanks to the next 8-character boundary.

Examples:

When you enter:

```
PRINT   /T
```

the computer empties the print queue.

When you enter:

```
PRINT   *.LST TEMP?.TST
```

all files with the extension .LST, and all files matching the file specification TEMP?.TST are printed.

When you enter:

```
PRINT   A:TEMP1.TST/C A:TEMP2.TST
A:TEMP3.TST
```

the system removes these three files from the print queue.

When you enter:

```
PRINT   TEMP1.TST/C TEMP2.TST/P
TEMP3.TST
```

the system removes TEMP1.TST from the print queue, and adds TEMP2.TST and TEMP3.TST.

When you enter:

```
PRINT   A:TEMP1.TST
```

the file TEMP1.TST is added to the print queue.

PROMPT Command

Type: Internal

Purpose: To change the MS-DOS command prompt.

Syntax: PROMPT [*prompt-text*]

Comments: This command changes the MS-DOS system prompt. If no parameters are entered, the screen displays the prompt for the default drive designator. To specify a special prompt, enter:

```
PROMPT $v
```

v may be one of the following special characters:

Character:	**Resulting Prompt:**
$	Dollar-sign ($) character
t	Current time
d	Current date
p	Current directory of the default drive
v	Version number
n	Default drive
g	Greater-than character (>)
l	Less-than character (<)
b	Bar (¦) character
_	CR LF sequence
h	Backspace plus deletion of previous character
e	ASCII code 1B (escape)
q	Equal sign (=)

You can create a prompt that begins with an MS-DOS command delimiter by preceding the delimiter with a special null character. A null character is any character not included in the previous list of special characters.

If you want to specify more than one character for the prompt, type a $ before each prompt character.

Examples:

Entering:

```
PROMPT   $n$g
```

sets the MS-DOS prompt (*d*>).

Entering:

```
PROMPT   Time=$t$_Date=$d
```

sets a two line prompt that displays:

```
Time = current time

Date = current date
```

Entering:

```
PROMPT   $z,HI
```

sets the MS-DOS prompt to:

```
,HI
```

Entering:

```
PROMPT   $p$g
```

sets the MS-DOS prompt to the default drive and working directory, and the greater-than sign, such as:

```
C:\ SUBDIR>
```

This is a useful format because it always indicates the working directory assignment.

RECOVER Command

Type:	External
Purpose:	To recover a file or an entire disk that has bad sectors.
Syntax:	RECOVER [*d*:][*pathname*]*filespec* or RECOVER *d*:

Comments: To recover a particular file, enter:

```
RECOVER [d:][pathname]filespec
```

MS-DOS then reads the file sector-by-sector, skipping the bad sectors. It finds and marks bad sectors, and allocates data to a different sector.

To recover a disk, enter the command and the drive designator. When you recover a disk, the recovered files are named FILEXXXX.REC. The XXXX in the new file specification stands for a numeric value that starts with 0001, and goes through 9999.

Only one file can be recovered when a file specification is used in the command. If wild card characters are used in the file specification, only the first file that matches the file specification is recovered. When a single file is recovered, the filename is not changed.

The size of the recovered file may be changed; the size of the recovered file is a multiple of the MS-DOS allocation unit size.

The files recovered on a disk can be examined to determine if they contain any useful data. If they do not, they can be erased.

When the recover command is executed, the following message is displayed:

```
Press any key to begin recovery of the
file(s) on drive d:
```

Press any key to continue with the recovery, or press ^BREAK to terminate the command.

Example:

RECOVER B:

If the root directory is full, the screen displays a message, and the computer stores information about the extra files in the file allocation table (FAT). The files may be moved to another disk and then erased from the original, in order to make room in the root directory. You can repeat the RECOVER command to put these files in the root directory when space becomes available.

REM (REMARK) Command

Type:	Internal (Batch)
Purpose:	To display a remark during the execution of a batch file.
Syntax:	REM [*comment*]
Comments:	The remark or comment is entered on the same line as REM, and is separated from the command by a blank space, tab, or comma. When the system sees REM in a batch file, it does not attempt to execute the remark that follows it.

REN (RENAME) Command

Type: Internal

Purpose: To rename a file.

Syntax: REN [*d:*][*pathname*]*filespec filespec*

or

RENAME [*d:*][*pathname*]*filespec filespec*

Comments: The first file specification has to be preceded by a drive designator, unless it resides on the disk in the default drive. Any drive designator for the second file specification is ignored. The renamed file continues to reside on the same disk.

Use of wild card characters causes a one-to-one correspondence between files specified in the first file specification and those specified in the second file specification.

The screen displays the error message "Duplicate file name or File not found" if you attempt to change a file specification to a name that already appears in the directory.

Examples:

```
REN  *.LST  *.PRN
```

All files with an .LST extension keep the same filename, but the extension changes to .PRN.

```
REN  B:ABODE  ?D?B?
```

Renames the file ABODE on the diskette in Drive B: to ADOBE.

RESTORE Command

Type: External

Purpose: To transfer files on a back-up diskette to a fixed disk.

Syntax: RESTORE *d:* [*d:*][*pathname*][*filespec*][/S][/P]

NOTE: You **must** include at least one of the following parameters: *d:*, *pathname*, or *filespec*.

Comments: Switches for RESTORE are:

/S Restores all back-up files matching *filespec* from the specified disk drive to the specified directory on the fixed disk, plus all subdirectories in that path. If you do not use the /S option, only the specified directory is restored.

/P Prompts before restoring read-only files, or files that have been modified since the last back-up.

RESTORE restores files to the fixed disk. Before the RESTORE command can be used, the files must have already been placed on the diskette with the BACKUP command.

The first drive specifier represents the drive for the diskette, and the second drive specifier represents the drive for the fixed disk. If you do not specify the name of the directory to be restored, the working directory is restored. If a pathname isn't specified, the system restores files to the current directory. If a file specification isn't given, all the files in the directory are restored.

The wild card characters ? and * can be used in the file specifications.

RESTORE sets EXIT codes as follows:

0 = Normal completion
1 = No files found
3 = Terminated by ^BREAK
4 = Terminated because of an error

When the RESTORE command is executed, the restore operation begins by displaying the following message:

```
Insert backup diskette 01 in drive d:

Strike any key when ready
```

Insert the correct diskette in the specified drive, and strike any key. The following messages are then displayed:

```
*** Files were backed up mm/dd/yyyy ***

*** Restoring files from diskette 01 ***
```

If a diskette that is not a back-up diskette is inserted, the following message is displayed:

```
Diskette is not a backup diskette
```

and the restore operation is terminated.

When restoring files from a series of diskettes, be sure to insert the diskettes in the proper sequence (it is important to label the diskettes during the back-up operation to maintain the correct sequence). If the diskettes are inserted in the wrong order, the following error message is displayed:

```
Warning ! Diskette is out of sequence
Replace the diskette or continue
Strike any key when ready
```

The correct diskette should then be inserted. Press any key to continue. As each file in the diskette is restored, the name of the file being restored is displayed. When all files on the diskette have been restored, RESTORE asks you to insert the next diskette in the series (if any). The restore operation completes when all specified files have been restored from all the diskettes in the series.

When restoring files to the fixed disk using the /S option, the first subdirectory to be restored must have been created on the fixed disk prior to the restore request. RESTORE creates all subsequent subdirectories in the specified path and restores the files matching the specified file specification to those subdirectories.

Examples:

Note

```
RESTORE   A:   C:\   /S
```

restores all files to the root directory, creates all subsequent subdirectories, and restores all files to those subdirectories on the fixed disk.

```
RESTORE  A:  C:\NOTES  /S
```

restores all files to the subdirectory NOTES, creates all subsequent subdirectories in that path, and restores all files to those subdirectories on the fixed disk. In this example, the subdirectory NOTES must already exist before you execute the RESTORE command.

```
RESTORE  A:  *.TXT  /S
```

restores all files with the .TXT extension to the working directory, and to all subdirectories in that path.

RMDIR (REMOVE DIRECTORY) Command

Type: Internal

Purpose: To remove a directory from a tree-structured directory.

Syntax: RMDIR [*d:*]*pathname*

or

RD [*d:*]*pathname*

Comments: Before removing a directory, you must delete all files contained in that directory. Use the DIR command with the pathname to view the list of files. Then use the DEL command to delete the files in that directory. The root directory (.) and parent directory (..) entries created by MS-DOS cannot be removed. Once the directory is empty, enter the command and pathname to delete the directory itself.

For example, to remove the directory \BIN\USER\SERVICE, first issue a DIR command for that path to list the files. Then delete all files listed for that directory. Then you can enter RMDIR and the pathname to remove the directory.

Example:

```
RMDIR  \BIN\USER\SERVICE
```

This deletes the entire directory named SERVICE (but not BIN or USER) from the directory structure on the disk in the default drive.

SET Command

Type: Internal

Purpose: To set one string value equal to another in the command processor's environment.

Syntax: SET [*string1* = [*string2*]]

Comments: Use this command to set string values that will later be used in an applications program or by an MS-DOS command. If SET is entered without parameters, all string values in the command processor's environment will be displayed.

If *string2* is omitted, *string1* is removed from the command processor's environment.

Lower-case letters in *string1* are replaced with upper-case letters, but *string2* is not changed.

The command processor's environment size is initially 127 bytes. When this area is full, it expands in increments of 127 bytes, if possible. Loading a program such as GRAPHICS or PRINT, which stays resident, prevents the area from expanding. The message "Out of environment space" is displayed if there is insufficient space for the new value.

SET is also used in batch processing to define the replaceable parameters with names instead of numbers.

Examples:

```
SET  TTY=VT52
```

This sets the TTY value to VT52, until it is changed with another SET command.

In a batch file, if there is a statement such as LINK %FILE%, the name MS-DOS uses for the variable can be set with the SET command. The command:

```
SET  FILE=DOMORE
```

replaces %FILE% with DOMORE.

SHELL Command

Type: Internal (Configuration)

Purpose: To give the name and location of a command processor other than COMMAND.COM that is used to load the system.

Syntax: SHELL =[*d:*][*pathname*]*filespec*

Comments: Use this configuration command to develop your own command processor.

This command must be used in your CONFIG.SYS file (see Section 4-7, HOW TO CONFIGURE YOUR SYSTEM).

SHIFT Command

Type:	Internal (Batch)
Purpose:	To access more than 10 replaceable parameters in a batch file.
Syntax:	SHIFT
Comments:	Command files usually have only 10 parameters (%0 through %9). When more than 10 parameters appear in a command line, the SHIFT command shifts each replaceable parameter up one, leaving off the first parameter. SHIFT reassigns the values %0 through %9 to the next ten parameters.

Example:

%0 = file
%1 = name
%2 = command
%3 = path
%4 = dog
%5 = cat
%6 = canary
%7 = mouse
%8 = goat
%9 = horse

Using the SHIFT command causes the following:

%0 = name
%1 = command
%2 = path
%3 = dog
%4 = cat
%5 = canary
%6 = mouse
%7 = goat
%8 = horse
%9 = unicorn

SORT Command

Type: External (Filter)

Purpose: To read, sort, and write data.

Syntax: SORT [/R][/+*n*]

Comments: Switches for SORT are:

/R	Reverses the sort, such as alphabetizing from Z to A.
/+*n*	Sorts columns beginning with the column number specified by *n*. If not entered, the sort begins with column 1.

SORT is a filter used to organize information. You must use redirection of input to sort data in a file.

The largest file that can be sorted is 63 Kbytes.

Examples:

```
SORT/R <UNSORT.TXT
```

Here, the system reads the file UNSORT.TXT, reserves the sort, and displays the results on the screen.

```
SORT <UNSORT.TXT    >>SORT.TXT
```

Here, the system reads the file UNSORT.TXT, sorts the data, and sends the output to the end of the file named SORT.TXT.

```
DIR¦SORT/+14
```

This command pipes the output of the directory command to the SORT filter, which sorts the directory listing starting with column 14, and then displays the output on the screen. Since column 14 lists the file size, the result is a directory sorted by file size.

```
DIR¦SORT/+14¦MORE
```

This command sorts the directory listing starting with column 14, and displays the output one screen at a time.

SYS (SYSTEM) Command

Type: External

Purpose: To transfer MS-DOS system files from one disk to another.

Syntax: SYS *d*:

Comments: SYS is normally used to place the two hidden operating system files (IBMBIO.COM and IBMDOS.COM) on a newly-formatted disk. A drive designator is required for the target disk.

Since the two system files are hidden files, they do not appear in the listing when you use the DIR command.

COMMAND.COM is not transferred with the SYS command.

Using the FORMAT command with the /S switch formats the disk and installs the system files on it.

SYS can be used to update an MS-DOS system diskette that contains an earlier version of DOS (such as version 1.XX). There must be enough diskette space available on the target diskette for the two hidden system files.

NOTE: Not all applications can be upgraded to version 2. Consult your dealer and/or application documentation to determine if your particular application will execute under MS-DOS VERSION 2.

TIME Command

Type: Internal

Purpose: To display and set the time.

Syntax: TIME [*hh*:*mm*[:*ss*[.*cc*]]]

Comments: Enter the TIME command in military hours. The abbreviations a.m. and p.m. are illegal. If TIME is entered without parameters, a message similar to the following one is displayed:

```
Current time is 13:32:15.08
Enter new time:
```

Type a new time, or press ENTER to bypass the prompt.

The new time must be entered using numerals only; letters are not allowed. The allowable parameters are:

hh = 00-24 (hours)
mm = 00-59 (minutes)
ss = 00-59 (seconds)
cc = 00-99 (hundredths of seconds)

The hour, minute, and second entries must be separated by colons. The hundredths-of-seconds entry is separated by a period. Seconds and hundredths of seconds are optional entries.

MS-DOS uses whatever time is entered as the new time, as long as the parameters and delimiters are legal. If the parameters or delimiters are illegal, MS-DOS returns the message:

```
Invalid time

Enter new time:
```

MS-DOS then waits for a legal time entry.

When you turn your computer off, the current time is cancelled. You must reset the time each time you turn your computer on, if you want to maintain the current time.

The system uses the current time in the file creation listing for the directory. When you display a directory listing, the time creation entry for files is displayed in conventional form, such as 1:30p.

TREE Command

Type: External

Purpose: To display all directory paths in a specified drive, and (optionally) to list all files in the working directory and its subdirectories.

Syntax: TREE [*d*:][/F]

Comments: The switch for TREE is:

/F Displays the names of the files in each subdirectory.

If you enter the TREE command without a specified drive, the system displays the full pathname of every directory and subdirectory on the disk in the default drive.

Example:

When you enter:

```
TREE B:
```

the screen displays something like this:

```
Path: \NEWDIR
Sub-directories: SUBDIR

Path: \NEWDIR\SUBDIR
Sub-directories: FILE
                 TRAINING

Path: \NEWDIR\SUBDIR\FILE
Sub-directories: None

Path: \NEWDIR\SUBDIR\TRAINING
Sub-directories: None
```

TYPE Command

Type: Internal

Purpose: To display the contents of a file.

Syntax: TYPE [*d:*][*pathname*]*filespec*

Comments: When you use TYPE to display the contents of a file, the symbols displayed are the input as the computer sees it. Files containing text are readable, but some files, such as program files, contain many unrecognizable characters.

NOTE:

- This command allows you to examine the file, but not modify it. Use EDLIN to change the contents.
- With the TYPE command, the contents of the file scroll up the screen very quickly. To temporarily suspend the display, press ^NUM LOCK. Press any character key to continue the display.
- To print the contents of the file, press ^PRTSC before you enter the TYPE command. The contents of the file print as they are displayed. To stop the printing, press ^PRTSC again.
- A display of binary files causes control characters, such as F6, to be sent to the system. This includes bells, form feeds, and escape sequences.

VER (VERSION) Command

Type:	Internal
Purpose:	To display the MS-DOS version number.
Syntax:	VER
Comments:	Type VER to display the MS-DOS version being used.

Example:

When you enter VER, the screen displays something similar to this:

```
COMPAQ Personal Computer MS-DOS Version 2.XX
```

VERIFY Command

Type:	Internal
Purpose:	To turn VERIFY on and off when writing to a disk.
Syntax:	VERIFY [ON|OFF]
Comments:	The VERIFY command and the /V switch in the COPY command perform the same task. VERIFY allows you to ensure that all data is written correctly to the disk. An error message appears if MS-DOS is unable to write data to the disk. VERIFY ON stays in effect until you change it in an applications program, or until you use VERIFY OFF to turn it off. To check the current setting of VERIFY, enter the command without parameters.

VOL (VOLUME) Command

Type: Internal

Purpose: To display the volume name, when one exists.

Syntax: VOL [*d*:]

Comments: This command prints the volume name of the disk in the specified drive. If you do not specify a drive, it prints the volume name of the disk in the default drive.

If the volume of the disk doesn't have an assigned label, this message appears:

```
Volume in drive d has no label
```

CHAPTER 5

EDLIN

5-1. INTRODUCTION TO EDLIN 5-1
5-2. RULES FOR USING EDLIN 5-3
5-3. EDLIN COMMANDS 5-5
  Intraline Editing 5-6
    Standard Editing Keys 5-6
    CTRL Key and Multiple-Key Functions 5-7
    Special Function Keys 5-8
  Interline Editing Commands 5-9
    Command Parameters 5-10
    APPEND (A) Command 5-13
    COPY (C) Command 5-14
    DELETE (D) Command 5-17
    EDIT Command 5-20
    END (E) Command 5-22
    INSERT (I) Command 5-24
    LIST (L) Command 5-29
    MOVE (M) Command 5-32
    PAGE (P) Command 5-34
    QUIT (Q) Command 5-37
    REPLACE (R) Command 5-38
    SEARCH (S) Command 5-43
    TRANSFER (T) Command 5-46
    WRITE (W) Command 5-48
  Special Function Keys 5-49
    F1 Key (Copy 1) 5-50
    F2 Key (Copy To) 5-51
    F3 Key (Copy All) 5-52
    DEL (DELETE) Key (Skip 1) 5-53
    F4 Key (Skip To) 5-54
    ESC (ESCAPE) Key (Skip All) 5-55
    INS (INSERT) Key (Add In) 5-56
    F5 Key (New Temp) 5-57
    F6 Key (^Z) 5-58

5-1. INTRODUCTION

EDLIN is a line editor used to edit, create, and display files one line at a time. Each line may be up to 253 characters long. The last character of each line is signaled when the ENTER key is pressed.

EDLIN is accessed by entering:

```
EDLIN   [d:] [pathname] [filespec] [/B]
```

If *filespec* is not given, this message appears:

```
File name must be specified
```

and the MS-DOS A> prompt appears. If you are creating the file, give it a name and press ENTER. EDLIN creates the file and returns the message "New file". The asterisk (*) prompt and the cursor appear, indicating that input may begin.

If the specified file already exists, EDLIN loads the file into memory. If the entire file is loaded, the screen displays the message "End of input file" and an asterisk (*) prompt. If the file is larger than memory, EDLIN fills three-fourths of available memory with the first part of the file and returns the asterisk (*) prompt, but not the "End of input file" message. The lines in the file may be called up individually by typing the line number, or listed using the LIST (L) command.

Line numbers are not present in saved text, but when a file is created or called up for edit, lines are numbered sequentially. Line numbers begin at 1 and are incremented by 1 through the end of the file. When new lines are inserted between existing lines, all line numbers following the inserted text are automatically increased by the number of lines inserted. When lines are deleted between existing lines, all line numbers following the deleted text are decreased automatically by the number of lines deleted. Consequently, line numbers always run sequentially from 1 through n (the number of the last line in the file). This is known as **dynamic numbering.**

If the file is longer than memory, and you want to edit the part that is not in memory, you must first write out to the disk some of the file that is in memory, using the WRITE (W) command, and then append more lines into memory using the APPEND (A) command.

To end the edit session, use the END (E) command to save the edit, or the QUIT (Q) command to quit the session without saving the edit. The Q command is especially useful if serious errors have been made during the editing session.

5-2. RULES FOR USING EDLIN

The following are rules for using EDLIN:

- Pathnames can be used as parameters. For example, if you enter:

  ```
  EDLIN \BIN\USER\SERVICE\TEXT.TXT
  ```

  you can use EDLIN to edit the TEXT.TXT file in the subdirectory SERVICE.

- Use a minus character (–) to refer to lines preceding the current line and a plus character (+) to refer to lines following the current line. For example:

  ```
  -10, + 10L
  ```

  lists the 10 lines preceding the current line, the current line, and the 10 lines following the current line.

- To issue multiple commands to edit one line, separate each command with a semicolon. If it is a REPLACE or SEARCH command, however, the string must be followed by a ^Z before the next command is given. For example:

  ```
  15;-5, + 5
  ```

  sets up line 15 for edit and displays lines 10 through 20.

  ```
  SThis string ^Z-5, + 5L
  ```

  searches for the string "This string", and displays the five lines before the match and the five lines following the match.

- Enter EDLIN commands with or without a space between the line number and the command. The commands 6 D and 6D, for example, are identical. In this manual, the commands will have a space between the line and the command.

- Insert control characters in the text by using ^V before an upper-case character. ^V instructs EDLIN to recognize the next character as a control character. Use ^V in any of the string arguments used in SEARCH and REPLACE commands. Some examples of ^V are shown following:

  ```
  S^VZ
  ```

  searches for the first occurrence of ^Z in a file.

  ```
  R^VZ^Zgoto
  ```

  replaces all occurrences of ^Z with "goto" in a file.

  ```
  R^VC^Zbar
  ```

  replaces all occurrences of ^C with "bar".

  Insert ^V in text by entering ^VV.

- To tell EDLIN to ignore ^Z in a file, use the /B switch.

- To edit files with a .BAK extension, rename the file using another extension, before starting EDLIN. In EDLIN, the system assumes that file specifications with a .BAK extension are back-up files and will not edit them.

5-3. EDLIN COMMANDS

EDLIN supports both interline and intraline commands. Interline commands are entered to perform editing functions on entire lines and files. Intraline commands use certain keys and key combinations to perform editing functions within a line.

The EDLIN interline editing commands are summarized below, and detailed later in this chapter.

APPEND (A)	Appends lines from the input file to the editing buffer.
COPY (C)	Copies a range of lines to a specified line number.
DELETE (D)	Deletes a specified line or line range.
EDIT	Calls up lines for editing.
END (E)	Saves an edited file on disk, creates a back-up file, exits the editing session, and returns to the MS-DOS command level.
INSERT (I)	Inserts line(s) of text in a newly-created file or immediately before a specified line.
LIST (L)	Displays a file or a specified range of lines on the screen.
MOVE (M)	Transfers a range of lines to a specified line.
PAGE (P)	Scrolls through a file 23 lines (approximately one "page", or screen) at a time.
QUIT (Q)	Stops the editing session without saving editing changes and exits to the MS-DOS command level.

REPLACE (R)	Replaces all occurrences of a specified string with a second string.
SEARCH (S)	Searches for a specified string.
TRANSFER (T)	Inserts the contents of a specified file into the file being edited at a specific line.
WRITE (W)	Writes lines from the editing buffer to an output file.

INTRALINE EDITING

Lines are edited in EDLIN through the use of special editing keys and functions keys that allow you to perform many frequently used commands with the touch of one or more keys. Only those keys that have specific applications in EDLIN are presented here.

Standard Editing Keys

The ENTER key is used to signal the end of an input line. In the insert mode, ENTER acts similarly to the return key on a typewriter. It accepts a line and moves the cursor down to the first position of the next line.

The SPACEBAR moves the cursor one space (left to right) each time it is pressed. If you space over a character, it is deleted.

BACKSPACE moves the cursor one space (right to left) each time it is pressed. If you backspace over a character, it is deleted.

The SHIFT keys on both sides of the keyboard change the keyboard output from lower-case letters to upper-case letters, or vice-versa. You always have to press SHIFT to type the symbols printed at the top of the keys.

CAPS LOCK changes the case of the alphabetic characters, but not the numbers and symbols. CAPS LOCK is a toggle key; press it to activate, and press it again to deactivate. If the alphabetic characters are in lower case, CAPS LOCK changes them to upper case. If they are upper case, CAPS LOCK makes them lower case. Whether or not CAPS LOCK is engaged, use SHIFT to type the opposite case for an alphabetic character, or to type one of the symbols printed at the top of a key.

TAB moves the cursor to the next tab stop. Tab stop default settings are every eight characters.

NUM LOCK is a toggle key. When it is on, the numbers in the number pad to the right of the keyboard are activated. When it is off, only the CURSOR-LEFT and CURSOR-RIGHT keys are operational. The CURSOR-LEFT key moves the cursor, deleting characters, from right to left. The CURSOR-RIGHT key enters the characters from the template to the buffer input line, one at a time, as does the special editing key F1 (see Special Function Keys). The CURSOR-UP, CURSOR-DOWN, HOME, END, PGUP, and PGDN keys have no function in EDLIN.

CTRL Key and Multiple-Key Functions

In this book, and in many program descriptions, there is a special symbol used to indicate control characters. For example, if the letter T is to be used as a control character, it is written as ^T. The caret (^) superscript always means that you must first press and hold down the CTRL key, and then press the designated character key. The ^T character, for example, is read aloud as "Control T".

If two keys are given (for example, ^BREAK), press and hold the first key, press the second key, and release them both. When three keys are given (for example, ^ALT+DEL), press and hold the first two (CTRL and ALT), press the third (DEL), and release all three.

Some CTRL key and multiple-key functions appear following:

^ALT + DEL resets the system to the MS-DOS command level.

^BREAK aborts the current command.

^NUM LOCK suspends display of output to the screen temporarily. This is especially useful if a large amount of material is being output to the screen, as when a very long directory is being displayed with the DIR command, or a long file is being displayed with the TYPE command. To continue, press any key.

^PRTSC activates the printer. Lines are printed as they are entered, or as they are displayed on the screen with the TYPE command. Press the keys again to deactivate the printer.

SHIFT + PRTSC causes the printer to print a hard copy of everything displayed on the screen, but it is not engaged on a continuous basis.

Special Function Keys

The EDLIN special function keys are summarized following, and detailed after the commands in this chapter.

Key:	**Function:**
F1 (Copy 1)	Copies one character from the template to the input buffer line. F1 is the same as the CURSOR-RIGHT key in the number pad.
F2 (Copy To)	Copies all characters, up to the character specified, from the template to the input buffer line.
F3 (Copy All)	Copies all, or the remaining characters from the template to the input buffer line.

Key:	Function:
DEL (Skip 1)	Skips over or deletes one character in the template line. To see the results of DEL, press F3. The remainder of the line is copied to the input buffer line.
F4 (Skip To)	Skips over characters in the template up to the character specified. Press F3 to copy and display the remainder of the line.
ESC (Skip All)	Voids the current input buffer line, leaving the template unchanged.
INS (Add In)	Turns the insert mode on and off. INS allows characters to be inserted into a line. INS is a toggle key; to cancel, press INS again. You can also cancel the insert mode by pressing ENTER, or any function key.
F5 (New Temp)	Moves the cursor down one line, and makes this line the new template for further editing.
F6 (^Z)	Ends the insert mode when you are writing lines to a file.

NOTE: Keys F7 through F10 have no functions assigned for EDLIN.

INTERLINE EDITING COMMANDS

Interline commands perform editing functions on entire lines or files. The interline commands are summarized below and described in detail following the description of command parameters.

Command Parameters

Interline commands, except END (E) and QUIT (Q), accept some parameters. The effect of a parameter depends on the command.

Parameter:		**Description:**
line		(Line number) A line number in a file. Line numbers must be separated from other line numbers by a comma. *line* may be specified in one of four ways:
	x	(Number) Any integer less than 65529. If a number larger than the largest existing line number is specified, *line* indicates the line after the last line number.
		(Period) If a period is specified for *line*, the period indicates the current line number. The current line is the most-recently-edited line, and not necessarily the last line displayed. The current line is marked by an asterisk (*) between the line number and the first character.
	#	(Pound sign) The # indicates the line after the last line number. Specifying # for *line* has the same effect as specifying a number larger than the last line number.
	↵	(ENTER key) Pressing the ENTER key, without any of the line specifiers above, directs EDLIN to use a default value appropriate to the command.
line,line		(Range of lines) Includes the first and last specified *line* and all lines in between. Line

Parameter:	**Description:**
	numbers are separated by a comma; for example 1,20 means lines 1 through 20. A single line number indicates the specified line only. A line number followed by a comma (for example 5,) specifies all lines from the given line through the end of the file. A line number followed by a comma and a period (for example 5,.) specifies all lines from the given line through the current line. If only the second line number is given, it must be preceded by a comma. This specifies all lines from the beginning of the file through the specified line. For example ,20 specifies lines 1 through 20. If the period is used to specify the current line, and the comma is used (for example .,20), all lines from the current line through line 20 are specified.
n	(Number of lines) Specifies a number of lines.
?	(Question mark) The question mark directs EDLIN to ask you if the correct string has been found. The question mark is used only with the REPLACE and SEARCH commands. Before continuing, EDLIN waits for a response. Use either Y or ENTER as a yes response, or any other key as a no response.
string	Specifies a string or group of characters that needs to be found or replaced, or that will

Continued

replace other text. *string* is used only with the SEARCH and REPLACE commands.

Each string must be terminated by ^Z or ENTER. No spaces should be left between strings, or between a string and its command letter, unless spaces are part of the string (see the SEARCH and REPLACE commands).

APPEND (A) Command

Purpose: To append lines from the input file to the editing buffer.

Syntax: [*n*] A

Comments: The APPEND command is used for larger files that will not fit into memory. Using the WRITE (W) command creates room for the file in memory. The APPEND (A) command is used to append lines to the file.

If A is typed without a parameter, lines are appended to that part of the file currently in memory until available memory is three-quarters full, or until there are no more lines to append.

If the parameter *n* is given, then *n* lines are appended to that part of the file currently in memory. The screen displays "End of input file" when the last line of the file has been added to memory.

COPY (C) Command

Purpose: To copy a range of lines to a specific line number.

Syntax: [*line*], [*line*], *line* [,*count*]C

Comments: If COPY is entered without the *count* parameter, the line is copied once. If the first or second *line* isn't given, the system defaults to the current line.

After the specified line is copied, the file is dynamically renumbered, including the copied line.

NOTE: If line numbers overlap, the message "Entry Error" appears.

Examples:

```
3,20,15 C
```

results in appearance of the "Entry Error" message.

```
1: This is a sample file
2: used to show copying lines.
3: See what happens when you use
4: the COPY command
5: (the C command)
6: to copy text in your file.
```

To copy this entire block of text once, enter:

```
1,6,7 C
```

To see the result, enter:

```
L
```

The LIST (L) command shows the file:

```
 1:  This is a sample file
 2:  used to show copying lines.
 3:  See what happens when you use
 4:  the COPY command
 5:  (the C command)
 6:  to copy text in your file.
 7:*This is a sample file
 8:  used to show copying lines.
 9:  See what happens when you use
10:  the COPY command
11:  (the C command)
12:  to copy text in your file.
```

To place lines within other text, the third *line* specifies the line before the inserted text. To insert text in the following file:

```
 1:  This is a sample file
 2:  used to show copying lines.
 3:  See what happens when you use
 4:  the COPY command
 5:  (the C command)
 6:  to copy text in your file.
 7:  You can also use COPY
 8:  to copy lines of text
 9:  to the middle of your file.
10:  End of sample file.
```

Enter:

```
3,6,10  C
```

The result is:

```
 1: This is a sample file
 2: used to show copying lines.
 3: See what happens when you use
 4: the COPY command
 5: (the C command)
 6: to copy text in your file.
 7: You can also use COPY
 8: to copy lines of text
 9: to the middle of your file.
10:*See what happens when you use
11: the COPY command
12: (the C command)
13: to copy text in your file.
14: End of sample file.
```

DELETE (D) Command

Purpose: To delete the specified line or line range.

Syntax: [*line*] [,*line*] D

Comments: If only one line number is given, the specified line is deleted.

If the line number is preceded by a comma, the second line number is assumed, and all lines from (and including) the current line up to (and including) the specified line are deleted.

If both lines are specified, separated by a comma, both lines and all lines in between are deleted.

A period indicates the current line and, when used with the comma, indicates the starting or ending line of the deletion.

NOTE: Use the DELETE (D) command with extreme caution. Lines and line ranges are permanently deleted. If you accidentally delete part of your file, use the QUIT (Q) command to abort the edit and start again.

When lines have been deleted, all line numbers following the deleted line or lines are dynamically renumbered. The line immediately after the deleted line or line range becomes the current line, and has the same line number as the first deleted line.

Example:

```
 1: This is a sample file
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: delete and insert
 .
 .
 .
25: (The D and I commands)
26: (Use ^Z to exit)
27:*line numbers.
```

To delete lines 5 through 24 enter:

```
5,24  D
```

The result is:

```
1: This is a sample file
2: used to demonstrate dynamic line numbers.
3: See what happens when you
4: delete and insert
5:*(the D and I commands)
6: (Use ^Z to exit)
7: line numbers.
```

To delete line 6, enter:

```
6 D
```

The result is:

```
1: This is a sample file
2: used to demonstrate dynamic line numbers.
3: See what happens when you
4: delete and insert
5: (the D and I commands)
6:*line numbers.
```

If you have a file with line 3 as the current line:

```
1:  This is a sample file
2:  used to demonstrate dynamic line numbers.
3:*See what happens when you
4:  delete and insert
5:  (the D and I commands)
6:  line numbers.
```

and you want to delete a range of lines beginning with the current line (3) and ending with line 5, enter:

```
.,5 D
```

or

```
,5D
```

The result is:

```
1:  This is a sample file
2:  used to demonstrate dynamic line numbers.
3:*line numbers.
```

EDIT Command

Purpose: To call up lines for editing.

Syntax: [*line*]

Comments: When a line number is entered, EDLIN displays the line number and the text for that line. This is the template. The line number is redisplayed on the line below, but the line is empty. This is the new input buffer line, and it is ready for editing.

Both lines show an asterisk (*) after the line number. The asterisk indicates that the line is the current line. In a listing, the current line is the last line entered. Any of the available intraline commands can be used to edit the line. The existing text serves as the template. When the line has been edited, or if no changes are needed, press the ENTER key to accept the line.

Each time you press ENTER, EDLIN alternates between presenting the prompt and cursor (*__), and the next line of the file. Each time the * prompt appears, EDLIN can accept a new command (LIST, END, etc).

If you press ENTER while the cursor is in the middle of a line, the remainder of the line is truncated (cut off).

NOTE: If the input buffer line needs additional editing, use the special editing keys to change the input buffer line. This may be repeated as often as necessary. When you press the ENTER key, the final input buffer line is sent into memory.

Example:

```
1:  This is a sample file
2:  used to demonstrate
3:  the editing of line
4:*four.
```

To edit line 4, enter:

```
4
```

The contents of the line are displayed with an asterisk on the template line. The input buffer line also has an asterisk. The cursor marks the first character space on the input buffer line.

```
4:*four.
4:*_
```

Now using the editing keys, enter:

```
INS (number  )     4:*number  _
F3                 4:*number four.
ENTER          *_
```

END (E) Command

Purpose: To save the edited file on disk, create a back-up file, exit the editing session, and return to the MS-DOS command level.

Syntax: E

Comments: To save the edited file on disk, enter E in response to the EDLIN prompt (*_). The system stores the file, renames the original file with a .BAK (back-up) extension, and exits to the MS-DOS command level. If the file was created during the editing session, a back-up file is not created.

When a file has been created using the INSERT (I) command, enter ^Z on the empty line following the last line of the insert. Then enter E at the EDLIN prompt.

NOTE: In the insert mode, ^Z causes the line where it is entered to be deleted. It should always be entered on an empty line below the last line of the file.

The E command takes no parameters.

If you did not select a drive when you started the editing session, the file system places the saved file on the default drive. You can use COPY to copy the file onto another disk. You must be sure, however, that the disk contains enough free space for the entire file to be written. If the disk does not contain enough free space, the write is aborted, and some or all of the edited file is lost.

After execution of the END (E) command, control is returned to the MS-DOS command level, and the screen displays the A> prompt.

To exit the editing session without saving the edit, use the QUIT (Q) command. You may want to use this command if you have made a serious editing error, such as entering D (DELETE) rather than L (LIST) after a line range. If the END (E) command is used, the lines are permanently deleted. The QUIT (Q) command voids all editing, including the errors, so you can edit the file again.

INSERT (I) Command

Purpose: To insert line(s) of text in a newly-created file or immediately before a specified line.

Syntax: [*line*] I

Comments: To create a file in EDLIN, enter:

```
EDLIN   filespec
```

EDLIN responds:

```
New file
*_
```

When you enter the INSERT (I) command, EDLIN sets up the first line for input:

```
*I
  1:*_
```

Each time ENTER is pressed, another line is added:

```
               1:*_
ENTER          2:*_
ENTER          3:*_
ENTER          4:*_
```

Examples:

```
1:  This is a sample file
2:  used to demonstrate dynamic line numbers.
3:  See what happens when you
4:  delete and insert
5:  (the D and I commands)
6:  (Use ^Z to exit)
7:*line numbers.
```

To insert text before a specific line (not the current line), enter a line number (for example 4) and I:

```
4 I
```

The result is:

```
4:*_
```

Enter the new text for lines 4 and 5:

```
4:*fool around with
5:*those very useful commands that
```

To end the insert, enter ^Z on line 6:

```
6:*^Z
```

NOTE: ^Z cancels and deletes the line where it is entered. Be sure to enter ^Z on an empty line following the last line in the file.

The LIST command (L) displays the file:

```
1: This is a sample file
2: used to demonstrate dynamic line numbers.
3: See what happens when you
4: fool around with
5: those very useful commands that
6:*delete and insert
7: (the D and I commands)
8: (Use ^Z to exit)
9: line numbers.
```

The lines you enter are inserted above line 4. When you have finished the insert, line 4 and all subsequent lines are automatically renumbered. The original line 4 is indicated as the current line, regardless of its new line number.

To insert lines immediately before the current line, enter:

```
I
```

The result is:

```
6:*_
```

Insert the following text and terminate with ^Z:

```
6:*perform the two major editing functions,
7:* ^Z
```

The result is:

```
 1: This is a sample file
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: fool around with
 5: those very useful commands that
 6: perform the two major editing functions,
 7:*delete and insert
 8: (the D and I commands)
 9: (Use ^Z to exit)
10: line numbers.
```

When the insert is finished, and you have exited the insert mode, the line immediately following the newly-inserted line(s) becomes the current line. All line numbers following the inserted section are dynamically renumbered.

To append lines to the end of a file, enter the number after the last line number before the INSERT (I) command. You can do this by entering the actual number after the last line (*n* I), or by entering # I, where # stands for the line number after the last line number in the file.

To append new lines to the end of the file, enter:

```
11 I
```

This produces:

```
11:*_
```

Enter the following new lines:

```
11:*The INSERT command can place new lines
12:*anywhere in the file.
13:*Because the line numbers are dynamic,
14:*they'll slide all the way to 65529.
```

End insertion by entering ^Z. The new lines appear at the end of all previous lines in the file.

The result is:

```
 1: This is a sample file
 2:  used to demonstrate dynamic line numbers.
 3:  See what happens when you
 4:  fool around with
 5:  those very useful commands that
 6:  perform the two major editing functions,
 7:  delete and insert
 8:  (the D and I commands)
 9:  (Use ^Z to exit)
10:  line numbers.
11:  The INSERT command can place new lines
12:  anywhere in the file.
13:  Because the line numbers are dynamic,
14:  they'll slide all the way to 65529.
```

Because you inserted lines to the end of the file, there is no asterisk (*) to indicate a current line.

If you do not specify an existing line, the default is the current line number.

If you specify an integer larger than the last line number, or if # is specified for *line*, the inserted lines are appended to the end of the file.

LIST (L) Command

Purpose: To display a file or a specified range of lines on the screen.

Syntax: [*line*] [,*line*] L

Comments: If L is entered without parameters, 23 lines are listed with the current line in the middle; that is, 11 lines before the current line, the current line, and 11 lines after the current line. The current line remains unchanged.

If both line numbers are given, separated by a comma, both lines and all lines between them are listed.

If only one line number is given, only that line is displayed.

If the line number is preceded by a comma, all lines preceding the specified line and including the specified line are displayed.

If the line number is followed by a comma, the specified line and the following 22 lines are displayed.

If a period is used to indicate the current line, the current line is used as the starting or ending line of the display.

Example:

```
 1: This is a sample file
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: delete and insert
 5: (the D and I commands)
 .
 .
 .
15:*The current line contains an asterisk.
 .
 .
 .
26: (Use ^Z to exit)
27: line numbers.
```

To list lines 2 through 5, enter:

```
2,5 L
```

The result is:

```
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: delete and insert
 5: (the D and I commands)
```

To list a range of lines beginning with the current line (15), enter:

```
.,26 L
```

The result is:

```
15:*The current line contains an asterisk.
 .
 .
 .
26: (Use ^Z to exit)
```

To list a range of 23 lines beginning with a specified line (13), enter:

```
13,  L
```

The result is:

```
13: The specified line is listed first in the range.
14: The current line remains unchanged.
15:*The current line contains an asterisk.
 .
 .
 .
35: ^Z exits interline INSERT command mode.
```

To list a range of 23 lines centered around the current line, enter:

```
L
```

The result is:

```
 4: delete and insert
 5: (the D and I commands)
 .
 .
 .
13: The current line is listed first in the range.
14: The current line remains unchanged.
15:*The current line contains an asterisk.
 .
 .
 .
26: (Use ^Z to exit)
```

MOVE (M) Command

Purpose: To transfer a range of lines to a specific line.

Syntax: [*line*], [*line*], *line* M

Comments: If M is entered without parameters, with invalid parameters, or if the line numbers overlap, the message "Entry error" appears.

If both optional *line* parameters are included, the specified lines, and all lines between them are moved immediately before the third specified line.

To move only one line, type the line number of the line to be moved, a comma, that same line number again, the number of the line you want it to appear before, and the command.

Examples:

If you enter:

```
20,30,100  M
```

the system takes the text from lines 20 through 30, and all lines in between, and moves them immediately before line 100.

If you enter:

```
6,6,10  M
```

the system moves the text on line 6 immediately before the text on line 10.

If you enter:

```
,+25,100  M
```

the system takes the text from the current line and the text from the following 25 lines and moves them before line 100.

If you enter:

```
9,+5,20  M
```

the system moves line 9, plus the following 5 lines, and places them before line 20.

If you enter:

```
,,10  M
```

the system moves only the current line, and places it before line 10.

Once the move is complete, the lines are dynamically renumbered.

PAGE (P) Command

Purpose: To scroll through a file 23 lines at a time or through a specified block of lines.

Syntax: [*line*] [,*line*] P

Comments: If P is entered without parameters, the system starts with the line following the current line, and lists the next 23 lines. The last line displayed becomes the current line, and is marked with an asterisk (*).

Examples:

If you enter:

```
P
```

with 35 being the current line, the system displays:

```
35:
36: The specified line is listed first.
37: The current line remains unchanged.
38: The current line contains an asterisk.
 .
 .
 .
57:*The current line contains an asterisk.
```

In this example, the system shows line 57 as the new current line.

If only the first *line* is specified, the system starts with the specified line, and lists the next 23 lines, marking the last line displayed with an asterisk.

If you enter:

```
24 P
```

the system displays:

```
24: This is a sample file
25: used to demonstrate dynamic line numbers.
26: See what happens when you
27: delete and insert
 .
 .
 .
43: This block of text
44: is being used to
45: demonstrate the MOVE command
46:*See what happens when you
```

If the beginning *line* and the second *line* are specified, the system starts with the first specified line, and lists all lines through the second specified line, regardless of the length of the block of lines. The last line displayed is marked with an asterisk (*). You cannot specify the second *line* parameter without specifying the first *line* parameter.

If you enter:

```
1,12   P
```

the system displays;

```
 1: This is a sample file
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: delete and insert
 5: (the D and I commands)
 6: This line is being moved to line x.
 .
 .
 .
12:*This block of text
```

QUIT (Q) Command

Purpose: To stop the editing session without saving editing changes, and exit to the MS-DOS command level.

Syntax: Q

Comments: The Q command takes no parameters. It is simply a fast way to exit an editing session without saving the edit. QUIT is especially useful if severe errors were made in the edit, such as typing D (DELETE), rather than L (LIST) after a line range. Q aborts the edit, mistakes and all.

When you want to stop the editing session, enter Q after the EDLIN prompt (*).

As soon as the QUIT command is given, EDLIN displays the message:

```
Abort edit   (Y/N)?
```

Press Y to stop the editing session; press N (or any character key) to restore the * prompt and continue the editing session.

To exit the editing session, save the edit, and make the original file a back-up file with a .BAK extension, use the END (E) command.

REPLACE (R) Command

Purpose: To replace all occurrences of the first string in the specified range of lines with the second string.

Syntax: [*line*] [,*line*] [?] R[*string1*] [^Z*string2*]

Comments: As each occurrence of *string1* is found, it is replaced by *string2*. Each line in which a replacement occurs is displayed. If a line contains more than one replacement, the line is displayed once for each occurrence.

Examples:

The following is a sample file:

```
 1:*This is a sample file
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: fool around with
 5: those very useful commands that
 6: perform the two major editing functions,
 7: delete and insert
 8: (the D and I commands)
 9: (Use ^Z to exit)
10: line numbers.
11: The INSERT command can place new lines
12: anywhere in the file.
13: Because the line numbers are sequential,
14: they'll slide all the way to 65529.
15: This is a sample file
16: used to demonstrate dynamic line numbers.
17: See what happens when you
18: fool around with
19: those very useful commands that
20: perform two major editing functions,
21: delete and insert
22: (the D and I commands)
23: (Use ^Z to exit)
```

If you enter:

```
2,14  Rand^Zor
```

the system displays:

```
  5: those very useful commors that
  7: delete or insert
  8: (the D or I commands)
  8: (the D or I commors)
 11: The INSERT commor can place new lines
```

If you omit the first *line* parameter, the system defaults to the line immediately following the current line, and ends with the second specified *line*. Therefore, if you enter:

```
,14  Rand^Zor
```

the system displays:

```
  5: those very useful commors that
  7: delete or insert
  8: (the D or I commands)
  8: (the D or I commors)
 11: The INSERT commor can place new lines
```

If you omit the second *line* parameter, the system defaults to the end of the file.

If you enter:

```
5, Rand^Zor
```

the system displays:

```
 5: those very useful commors that
 7: delete or insert
 8: (the D or I commands)
 8: (the D or I commors)
11: The INSERT commor can place new lines
19: those very useful commors that
21: delete or insert
22: (the D or I commands)
22: (the D or I commors)
```

If a question mark (?) is entered after the *line* parameters, the system prompts you with an "O.K.?" so that you can verify that you want to make the change before the system replaces the string. If you answer Y (for yes) to the prompt, the system replaces the specified string and that line becomes the current line. If you answer N (for no), the system advances to the next line containing the specified string and prompts you again.

If you omit both *line* parameters, the system defaults to the line immediately following the current line, and goes to the end of the file.

For example, if your current line is 6, and you enter:

```
? Rand^Zor
```

the system displays something like this:

```
     7: delete or insert
O.K.? y
```

```
    8: (the D or I commands)
O.K.? y
    8: (the D or I commors)
O.K.? n
   11: The INSERT commor can place new lines
O.K.? n
   19: those very useful commors that
O.K.? n
   21: delete or insert
O.K.? y
   22: (the D or I commands)
O.K.? y
   22: (the D or I commors)
O.K.? n
   25: The INSERT commor can place new lines
O.K.? n
```

If you omit *string1*, the system defaults to the last-used *string1*.

For example, if the previously-entered command (1,14, Rand^Zor) caused the system to display:

```
  5:  those very useful commors that
  7:  delete or insert
  8:  (the D or I commands)
  8:  (the D or I commors)
 11:  The INSERT commor can place new lines
```

and you enter:

```
15,28   R^Zor
```

the system displays something like this:

```
19:  those very useful commands that
21:  delete and insert
22:  (the D or I commands)
22:  (the D or I commors)
25:  the INSERT commor can place new lines
```

If you omit *string2*, the system assumes that you do not want to replace *string1* with anything and deletes all occurrence of *string1*.

For example, if you enter:

```
1,  Rand^Z
```

the system displays:

```
 5:  those very useful comms that
 7:  delete insert
 8:  (the D I commands)
 8:  (the D I comms)
11:  The INSERT comm can place new lines
19:  those very useful comms that
21:  delete insert
22:  (the D I commands)
22:  (the D I comms)
25:  The INSERT comm can place new lines
```

If you omit both *string1* and *string2*, the system defaults to the last-used *string1*, and deletes all occurrences of it.

For example, if you entered:

```
1,14  Rand^Z
```

and then you enter:

```
15,28   R
```

the system displays:

```
19:  those very useful commors that
21:  delete or insert
22:  (the D or I commands)
22:  (the D or I commors)
25:  The INSERT commor can place new lines
```

SEARCH (S) Command

Purpose: To search the specified ranges of lines for the specified string.

Syntax: [*line*] [,*line*] [?] S*string*

Comments: *string* must be terminated with ENTER. The first line that matches *string* is displayed and becomes the current line. The SEARCH (S) command terminates when a match is found. If no line contains a match for *string*, the message "Not found" is displayed.

If the optional parameter ? is included in the command, EDLIN displays the first line with a matching *string*, and then prompts you with the message "O.K.?". If you press either Y or ENTER, the line becomes the current line, and the search terminates. If you press any other key, the search continues until another match is found, or until all lines have been searched; then the "Not found" message is displayed.

If the first *line* is omitted (as in ,*line* S*string*), the first *line* defaults to the line after the current line. If the second *line* is omitted (as in *line* S*string*), the second *line* defaults to # (line after the last line), which is the same as *line*,# S*string*.

If *string* is omitted, no search is made and the message "Not found" appears.

Examples:

```
1: This is a sample file
2: used to demonstrate dynamic line numbers.
3: See what happens when you
```

```
 4: fool around with
 5: those very useful commands that
 6: perform the two major editing functions,
 7: delete and insert
 8: (the D and I commands)
 9: (Use ^Z to exit)
10: line numbers
11: The INSERT command can place new lines
12: anywhere in the file.
13: Because the line numbers are dynamic,
14:*they'll slide all the way to 65529.
```

To search for the first occurrence of the string "and", enter:

```
2,12  Sand
```

The result is:

```
5:*those very useful commands that
```

To get to the "and" in line 7, modify the SEARCH (S) command by entering:

```
,12 Sand (space)
```

The search begins from the line after the current line (5), since no first line is given. The result is:

```
7:  delete and insert
```

To search through several occurrences of a string until the correct string is found, enter:

```
1,?  Sand
```

The result is:

```
  5:*those very useful commands that
O.K.?_
```

If any key except Y or ENTER is pressed, the search continues. Type N to continue:

```
  7:  delete and insert
O.K.?_
```

Press Y to terminate the search:

```
O.K.? Y
*_
```

TRANSFER (T) Command

Purpose: To insert the contents of a specified file into the file that is being edited, beginning at the specific line.

Syntax: [*line*] T*filespec*

Comments: If the line number is omitted, the contents of the file are inserted at the current line. Following the transfer, all lines are dynamically renumbered.

Example:

If you have a file:

```
 1: This is a sample file
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: fool around with
 5: those very useful commands that
 6: perform the two major editing functions,
 7: delete and insert
 8: (the D and I commands)
 9: (Use ^Z to exit)
10: line numbers.
11:
12:
13: The INSERT command can place new lines
14: anywhere in the file.
15: Because the line numbers are dynamic,
16: they'll slide all the way to 65529.
```

and you enter:

```
12, Tsample
```

The result is:

```
 1: This is a sample file
 2: used to demonstrate dynamic line numbers.
 3: See what happens when you
 4: fool around with
 5: those very useful commands that
 6: perform the two major editing functions,
 7: delete and insert
 8: (the D and I commands)
 9: (Use ^Z to exit)
10: line numbers.
11:
12:*This is a sample file
13: used to demonstrate
14: the editing of line
15: number four.
16:
17: The INSERT command can place new lines
18: anywhere in the file.
19: Because the line numbers are dynamic,
20: they'll slide all the way to 65529.
```

WRITE (W) Command

Purpose: To write lines from the editing buffer to the output file.

Syntax: [*n*] W

Comments: The WRITE command is used when editing files that are larger than available memory. By executing WRITE, lines are written out to the output file, and room is made in the editing buffer for more lines to be appended from the input file.

If W is typed without the *n* (number of lines) parameter, lines are written out until memory is three-quarters full.

If the *n* parameter is given, *n* lines are written out. Lines are written out beginning with the start of the file (line 1). Subsequent lines in the editing buffer are renumbered beginning with 1. A later APPEND (A) command will append any remaining lines to the editing buffer.

SPECIAL FUNCTION KEYS

These special function keys perform specific editing tasks with the touch of a single key. A description of the special function keys follow.

F1 Key (Copy 1)

Purpose: To copy one character from the template to the input buffer line.

Comments: At the beginning of the intraline edit, the cursor (indicated by the underline) is positioned at the beginning of the input buffer line.

Pressing the F1 key copies one character from the template (the last-entered line) to the input buffer line (line where the cursor is positioned). Insert is automatically turned off, if it was on. If ENTER is pressed before the complete line has been copied, the template will be truncated from the point of entry.

NOTE: The CURSOR-RIGHT key in the number pad has the same function.

Example:

```
1:*This is a test file
1:*_
```

F1 copies the first character (T) from the template to the second, or input buffer line:

```
             1:*This is a test file
F1           1:*T_
```

Each time F1 is pressed, one more character appears:

```
F1           1:*Th_
F1           1:*Thi_
F1           1:*This_
```

F2 Key (Copy To)

Purpose: To copy all characters up to the first occurrence of a specified character from the template to the input buffer line.

Comments: Pressing the F2 key copies all characters up to, but not including, a specified character. The specified character is not copied or displayed on the screen.

Using the SPACEBAR to enter a space for the specified character causes all characters up to the first space to be copied. If the template does not contain the specified character, nothing is copied. F2 automatically turns insert off, if it is on.

NOTE: Because F2 does not copy the character specified, F2 cannot be used to copy the last character of a line from the template to the input buffer line. Use F1 or F3 to complete the line. If ENTER is pressed before the buffer line is complete, the template will be truncated from the point of entry.

Example:

To copy all characters up to the character E:

```
            1:*This is a test file
F2e         1:*This is a t_
```

F3 Key (Copy All)

Purpose: To copy the entire line, or the remainder of the line, from the template to the input buffer line.

Comments: Pressing F3 copies all characters from the template, or copies all those remaining from the last character edited. When F3 is pressed, the rest of the line appears on the buffer line, and the cursor is positioned after the last character on the line. Insert is automatically turned off, if it is on.

Only F1 or F3 will print the last character of the template to the input buffer line. If ENTER is pressed before the last character of the template is copied, the template will be truncated from the point of entry.

Example:

```
        1:*This is a test file
F3      1:*This is a test file_
```

DEL (DELETE) Key (Skip 1)

Purpose: To skip (not copy) one character from the template to the buffer line.

Comments: Each time DEL is pressed, one character is skipped (deleted) from the template. The action of DEL is opposite to that of F1. When the edited buffer line is entered, those characters not copied from the template are effectively deleted.

Example:

```
                1:*This is a test file
    DEL         1:*_
```

The cursor does not move. Press DEL as many times as needed to delete the appropriate characters. To see how much of the line has been skipped, press F3 to copy the rest of the template to the buffer line.

```
                1:*This is a test file
    DEL         1:*_
    F3          1:*his is a test file_
```

F4 Key (Skip To)

Purpose: To skip (not copy) multiple characters up to, but not including, a specified character from the template to the buffer line.

Comments: Pressing F4 skips all characters in the template up to the first occurrence of a specified character. The specified character is not copied or displayed on the screen.

To see the results of F4, press F3 to copy the rest of the template to the buffer line. The specified character is the first character displayed.

If the template does not contain the specified character, nothing is skipped. The action of F4 is opposite to that of F2.

Example:

To skip all characters up to the character E:

```
          1:*This is a test file
F4e       1:*_
F3        1:*est file_
```

ESC (ESCAPE) Key (Skip All)

Purpose: To stop inputting and cancel the input buffer line.

Comments: Pressing ESC cancels the input buffer line, but leaves the template unchanged. ESC displays a backslash (\) at the end of the line you cancelled, moves the cursor to the first position of the next line, and turns the insert mode off, if it was on.

Pressing F3 copies the template to the input buffer line just as it was before ESC was pressed.

Example:

```
            1:*This is a test file
F4e         1:*_
F3          1:*est file
ESC         1:*est file\
               _
F3             This is a test file_
```

INS (INSERT) Key (Add In)

Purpose: To insert (not copy over) characters into a line.

Comments: Pressing INS turns the insert mode on. Characters are inserted to the left of the character the cursor is under. Characters to the right are automatically adjusted to make room for the insert. Pressing INS again cancels the insert mode. You can also cancel the insert mode by pressing ENTER, or any function key.

NOTE: If you press the ENTER key before copying or inserting the rest of the line, the line will be truncated at the point of entry.

Example:

```
                 1:*This is a test file
   F2t           1:*This is a __
```

Press INS to insert the word "brief", followed by a space:

```
   INS (brief  ) 1: This is a brief __
```

Press F3 to copy the rest of the template:

```
   F3            1:*This is a brief test file__
```

If you press ENTER, rather than F3, the remainder of the template is truncated. The input buffer line ends after the insert:

```
                 1:*This is a brief
   ENTER         2:*__
```

F5 Key (New Temp)

Purpose: To make the edited input buffer line the new template for additional editing.

Comments: Pressing F5 makes the edited input buffer line the new template to allow further editing. Pressing F5, rather than ENTER, at the end of the edited line displays an at sign (@) at the end of the line, and positions the cursor, without the * prompt, at the first character of the next line. A new line number is not shown.

Example:

```
                1:*This is a test file
F2t             1:*This is a _
INS (brief   )  1:*This is a brief _
F3              1:*This is a brief test file
F5              1:*This is a brief test file@
                   _
```

Additional editing can now be done using the new template line.

To delete the word "brief" and return the input buffer line to its original form, press:

```
F5              1:*This is a brief test file@
                   _
F2b             1:*This is a _
F4t             1:*This is a _
F3              1:*This is a test file_
```

F6 Key (^Z)

Purpose: To end the insert mode

Comments: F6 is the same as ^Z.

When you are writing lines to a file, pressing F6 terminates the insert mode. Go to the next blank line following the end of the file and press F6. The ^Z symbol appears on the current line. When you press ENTER, the EDLIN prompt (*__) returns, and you can enter another EDLIN command.

CHAPTER 6

MS-LINK

6-1. OVERVIEW 6-1
Capabilities of the Linker Utility 6-1
MS-LINK Operation Summary 6-1
Definitions 6-3
6-2. COMBINING AND ARRANGING SEGMENTS 6-5
6-3. FILES USED BY MS-LINK 6-9
MS-LINK Input Files 6-9
MS-LINK Output Files 6-9
The VM.TMP File 6-10
6-4. RUNNING MS-LINK 6-11
Command Prompts 6-11
Switches 6-13
/DSALLOCATE 6-14
/HIGH 6-14
/LINENUMBERS 6-15
/MAP 6-15
/PAUSE 6-15
/STACK:*number* 6-16
Command Characters 6-16
Invoking MS-LINK 6-18

6-1. OVERVIEW

CAPABILITIES OF THE LINKER UTILITY

The MS-LINK utility allows you to link object modules and produce object code that can be relocated and executed.

MS-LINK is a virtual linker, which means that it can link programs that are too large for the available memory. This is possible because the output file from MS-LINK (run, or .EXE, file) is not bound to specific memory addresses; therefore, it can be loaded and executed at any convenient address by the operating system. MS-LINK is capable of linking files totalling 384 Kbytes.

MS-LINK automatically prompts you for the names of input and output modules and other MS-LINK parameters, and handles user-defined overlays. MS-LINK can also use an automatic response file to answer these prompts.

MS-LINK uses a dictionary-indexed library search method, which substantially reduces linking time for sessions involving library searches.

MS-LINK OPERATION SUMMARY

Programs are initially written in source code. The source code must then be passed through a compiler, which produces object modules. Object modules are programs that are in a form that the computer can understand directly (machine language). You may wish to combine (link) several programs to be run at the same time. The MS-LINK utility allows you to combine several object modules into one relocatable load module, or run file (called an .EXE, or executable, file).

Each program may refer to symbols that are defined in other object modules. These are called external references. As it combines modules, MS-LINK defines (resolves) external references between object modules and can search several library files for definitions of external references not defined in the object modules. MS-LINK produces a list file of external references that have been resolved, and any error messages.

MS-LINK uses available memory as much as possible. When available memory is exhausted, MS-LINK creates a temporary disk file (VM.TMP) for the additional data, becoming a virtual linker.

A block diagram of MS-LINK operation is shown in Figure 6-1.

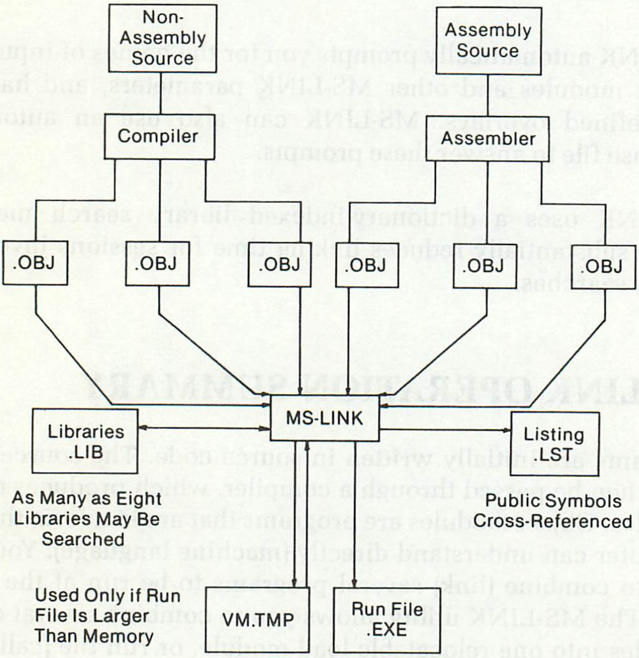


Figure 6-1. MS-LINK Operation Block Diagram

DEFINITIONS

Some of the terms used in this chapter are explained in the following paragraphs to help you understand how MS-LINK works. This information is needed mainly if you are writing and compiling programs in assembly language.

MS-DOS divides memory into **segments, classes,** and **groups.**

For example:

	Segment Name:	**Class Name:**
Segment 1	PROG.1	CODE
Segment 2	PROG.2	CODE
Segment 12	PROG.3	DATA

In this example, segments 1, 2, and 12 have different segment names, but may or may not have the same class name. Segments 1, 2, and 12 form a group with a group address of the lowest address of segment 1 (the lowest address in memory).

Each segment has a **segment name** and a **class name.** MS-LINK loads all segments into memory, ordered by class name, from the first segment encountered to the last. All segments assigned to the same class are loaded into memory in sequence, or contiguously.

During processing, MS-LINK references segments by their addresses in memory (where they are located). MS-LINK does this by finding groups of segments.

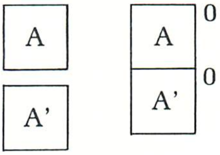
A **group** is a collection of segments that fit within a 64-Kbyte area of memory. They do not need to be contiguous to form a group. The address of any group is the lowest address of the segments in that group. At link time, MS-LINK analyzes the groups, then references the segments by the address in memory of that group. A program may consist of one or more groups.

If you are writing in assembly language, you can assign the group and class names in your program. In high-level languages (BASIC, COBOL, FORTRAN, and Pascal), the naming is done automatically by the compiler.

6-2. COMBINING AND ARRANGING SEGMENTS

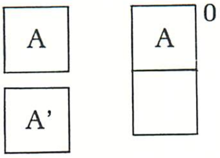
MS-LINK works with four combine types, which are declared in the source module for the assembler or compiler. These combine types are private, public, stack, and common. MS-LINK combines segments for these combine types as follows:

Private:



Private segments are loaded separately and remain separate. They may be physically contiguous, but not logically, even if the segments have the same name. Each private segment has its own base address.

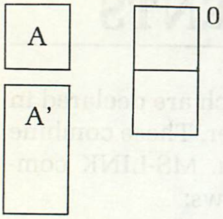
Public/Stack:



Public segments of the same segment name and class name are loaded contiguously. Offset is from the beginning of first segment loaded through the last segment loaded. There is only one base address for all public segments of the same segment name and class name.

The combine types stack and memory are treated the same as public; however, the stack pointer (SP) is set to the first address of the first stack segment.

Common:



Common segments of the same segment name and class name are loaded, overlapping one another. There is only one base address for all common segments of the same name. The length of the common area is the length of the longest segment.

Placing segments in a group in the assembler provides offset addressing of items from a single base address for all segments in that group (see Figure 6-2).

For example:

```
A   SEGMENT  'CODE'
A   ENDS
B   SEGMENT  'CONST'
B   ENDS
C   SEGMENT  'DATA'
C   ENDS
D   SEGMENT  STACK  'STACK'
D   ENDS
E   SEGMENT  'MEMORY'
E   ENDS
```

To have memory combine type to be loaded as the last segments of your program, you can simply add MEMORY between SEGMENT and 'MEMORY' in the preceding E segment line. Note, however, that these segments are loaded last only because you imposed this control on them, and not because of any inherent capability in the linker or assembler operations.

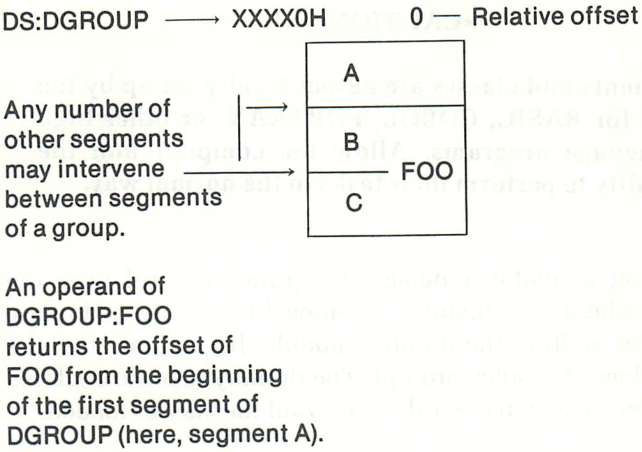

DS:DGROUP → XXXX0H
0 — Relative offset
A
B
C
FOO
Any number of other segments may intervene between segments of a group.
An operand of DGROUP:FOO returns the offset of FOO from the beginning of the first segment of DGROUP (here, segment A).

Figure 6-2. Offset Addressing of Items From a Single Base Address

Segments are grouped by declared class names. MS-LINK loads all the segments belonging to the first class name encountered, then loads all the segments of the next class name encountered, and so on until all classes have been loaded. For example:

If your program contains:

```
A  SEGMENT  'FOO'
B  SEGMENT  'BAZ'
C  SEGMENT  'BAZ'
D  SEGMENT  'ZOO'
E  SEGMENT  'FOO'
```

They will be linked in the following order:

All segments named 'FOO' — A
E
All segments named 'BAZ' — B
C
All segments named 'ZOO' — D

CAUTION

The segments and classes are automatically set up by the compiler for BASIC, COBOL, FORTRAN, or other high level language programs. Allow the compiler and the linker utility to perform their tasks in the normal way.

When writing assembly language programs, control over the ordering of classes in memory is done by writing a dummy module. After writing the dummy module, list it first after the MS-LINK Object Modules prompt. The dummy module declares segments into classes in the order you want the classes loaded.

Caution should be taken to declare all classes to be used in your program in this module. If you do not, you lose absolute control over the ordering of classes.

6-3. FILES USED BY MS-LINK

MS-LINK uses one or more input files to produce two output files. Additionally, MS-LINK may create a virtual memory file, and may be directed to search one to eight library files. For each file, you may assign a file specification. The format for MS-LINK file specifications is:

[d:]*filename.ext*

MS-LINK INPUT FILES

If no extensions are given as part of the input (object) file specifications, MS-LINK recognizes by default:

File:	**Default Extension:**
Object	.OBJ
Library	.LIB

MS-LINK OUTPUT FILES

MS-LINK appends to the output (run and list) files the following default extensions:

File:	**Default Extension:**
Run	.EXE (may not be overridden)
List	.MAP (may be overridden)

THE VM.TMP FILE

MS-LINK uses available memory for the linking session. If the files to be linked create an output file that exceeds the available memory, MS-LINK creates a temporary file and names it VM.TMP. If MS-LINK needs to create VM.TMP, it displays the message:

```
VM.TMP has been created
Do not change disk in drive (A: or B:)
```

Once this message is displayed, you must not remove the diskette from the default drive until the link session ends. If the diskette is removed, the operation of MS-LINK is unpredictable, and MS-LINK might return the error message:

```
Unexpected end of file on VM.TMP
```

MS-LINK uses VM.TMP as virtual memory. The contents of VM.TMP are subsequently written to the file named following the Run File: prompt. VM.TMP is a temporary working file, and is deleted at the end of the linking session.

CAUTION

Do not use VM.TMP as a file specification for any other file. If you have a file named VM.TMP on the disk in the default drive and MS-LINK requires the VM.TMP file, MS-LINK deletes the VM.TMP on disk and creates a new VM.TMP. The contents of the previous VM.TMP file is lost.

6-4. RUNNING MS-LINK

Running MS-LINK requires two types of input: a command to start MS-LINK and responses to the command prompts. There are also six switches that control MS-LINK features. The command prompts appear one after another as you answer each one. When the last prompt has been answered, MS-LINK begins to run automatically. When the linking session is finished, MS-LINK exits to the operating system. If the linking session is unsuccessful, MS-LINK returns the appropriate error message.

Commands to control MS-LINK are usually entered from the keyboard, but responses to the command prompts and the switches you want to use can be contained in a response file or on the command line.

COMMAND PROMPTS

MS-LINK prompts you for the names of object, run, list, and library files. These prompts are described in the following list in their order of appearance. For prompts which can default to preset responses, the default response is shown in square brackets ([ ]) following the prompt.

Object Modules [.OBJ]:

The Object Modules: prompt is followed by only an extension default response (.OBJ) because it has no preset filename response, and requires you to supply a filename.

If MS-LINK cannot find an object module in a disk drive, it displays the message:

```
Cannot find file <filename>
Change diskette <hit ENTER>
```

and allows you to change diskettes.

If MS-LINK looks on the fixed disk, or on a diskette where VM.TMP is open, and cannot find an object module, it displays the message:

```
Cannot find file filespec
```

and terminates.

List the object modules to be linked. They must be separated by blanks or plus signs (+). If a plus sign is the last character typed, the prompt reappears. There is no default; a response is required.

Remember that MS-LINK loads segments into classes in the order encountered. Use this information for setting the order in which the object modules are entered.

Run File [Object-file.EXE]:

Supply a filename for the executable object code. The default is first-object-filename.EXE (the output extension cannot be changed).

List File [NUL.MAP]:

Supply a filename for the listing. The default is NUL.MAP (the output extension can be changed). If the list file goes to a diskette, MS-LINK will not allow you to remove that diskette during the linking process.

Libraries [.LIB]

The valid responses are zero to eight library filenames. The library files must have been created by a library utility. MS-LINK assumes, by default, that the extension is .LIB for the library files.

Library filenames must be separated by blank spaces or plus signs (+).

MS-LINK searches the library files in the order listed to resolve external references. When it finds the module that defines the external symbol, MS-LINK processes the module as another object module.

If MS-LINK cannot find a library file on the disk, it returns the message:

```
Cannot find library library-name
Enter new drive letter:
```

Simply enter the letter for the drive designator (for example, A).

MS-LINK does not search within each library file sequentially. MS-LINK uses a method called dictionary-indexed library search. This means that it finds definitions for external references by index access rather than searching from the beginning of the file to the end for each reference. This indexed search substantially reduces the link time for any sessions involving library searches.

SWITCHES

There are six switches that control alternate linker functions. Switches must be entered at the end of a prompt response, regardless of which method is used to invoke MS-LINK. Switches may be grouped at the end of any response, or may be scattered at the end of several. If more than one switch is entered at the end of any one response, each switch must be preceded by a slash (/).

Abbreviations can be entered for any switch, from a single letter through the whole switch name. The only restriction is that an abbreviation must be a sequential substring from the first letter through the last letter of the switch name; no gaps or transpositions are allowed. For example, to enter the /DSALLOCATE switch:

Legal:	**Illegal:**
/D	/DSL
/DS	/DAL
/DSA	/DLC
/DSALLOCA	/DSALLOCT

The switches are described in the following list.

/DSALLOCATE

The /DSALLOCATE switch directs MS-LINK to load all data (DGROUP) at the high end of the data segment (DS). Otherwise, MS-LINK loads all data at the low end of DS. At runtime, the DS pointer is set to the lowest possible address and allows the entire DS segment to be used. If the /DSALLOCATE switch is used in combination with the default load low (that is, the /HIGH switch is not used), you can dynamically allocate any available memory below the area specifically allocated within DGROUP. The memory remains addressable by the same DS pointer. This dynamic allocation is needed for Pascal and FORTRAN programs.

NOTE: Your applications program may dynamically allocate up to 64 Kbytes (or the actual amount available), less the amount allocated within DGROUP.

/HIGH

Use of the /HIGH switch causes MS-LINK to place the run file as high as possible in memory. Otherwise, MS-LINK places the run file as low as possible.

CAUTION

Do not use the /HIGH switch with Pascal or FORTRAN programs.

/LINENUMBERS

Use of the /LINENUMBERS switch directs MS-LINK to include in the list file the line numbers and addresses of the source statements in the input modules. Otherwise, line numbers are not included in the list file.

NOTE: Not all compilers produce object modules that contain line number information. In these cases, MS-LINK cannot include line numbers.

/MAP

The /MAP switch directs MS-LINK to list all public (global) symbols defined in the input modules. If /MAP is not given, MS-LINK lists only errors (which includes undefined globals).

The symbols are listed alphabetically. For each symbol, MS-LINK lists its value and its segment:offset location in the run file. The symbols are listed at the end of the list file.

/PAUSE

The /PAUSE switch causes MS-LINK to pause in the link session. Normally, MS-LINK performs the linking session without stopping. The /PAUSE switch allows you to exchange the diskettes before MS-LINK outputs the run (.EXE) file.

When MS-LINK encounters the /PAUSE switch, it displays the message:

```
About to generate .EXE file
Change disks hit any key
```

MS-LINK resumes processing when you press any key.

CAUTION

> **Do not exchange the diskette which will receive the list file, or the diskette used for the VM.TMP file, if it has been created.**

/STACK:*number*

number represents any positive numeric value up to 65536 bytes. If the /STACK switch is not used for a link session, MS-LINK calculates the necessary stack size automatically.

If a value from 1 to 511 is entered, MS-LINK uses 512.

All compilers and assemblers should provide information in the object modules that allow MS-LINK to compute the required stack size. At least one object (input) module must contain a stack allocation statement. If not, MS-LINK returns the message:

```
WARNING: NO STACK STATEMENT
```

COMMAND CHARACTERS

MS-LINK provides the following three command characters:

+ Use the plus sign (+) to separate entries and to extend the current physical line following the Object Modules: and Libraries: prompts. A blank space may be used to separate object modules. To enter a large number of responses (each of which may also be very long), enter a plus sign at the end of the physical line to extend the logical line. If the plus sign is the last entry following these two prompts, MS-LINK prompts you for more module names.

When the Object Modules: or Libraries: prompt appears again, continue to enter responses. When all the modules to be linked have been listed, verify that the response line ends with a module name, not a plus sign.

Examples:

```
Object Modules [.OBJ]: FUN TEXT TABLE CARE
Object Modules [.OBJ]: FOO+FLIPFLOP+ JUNQUE+
Object Modules [.OBJ]: CORSAIR
```

; To save time, you can use a single semicolon (;) at any time after the first prompt (from Run File: on) to select default responses to the remaining prompts.

The remaining prompts do not appear, and MS-LINK uses the default values (including FUN.MAP for the list file).

NOTE: Once the semicolon has been entered, you can no longer respond to any of the prompts for that MS-LINK session. Therefore, you cannot use the semicolon to skip over only some prompts.

Example:

```
Object Modules [.OBJ]: FUN TEXT TABLE CARE
Run File [FUN.EXE]:;
```

^BREAK Use ^BREAK at any time you want to abort the link session. If you enter an incorrect response, such as the wrong filename, or misspelled filename, you must press ^BREAK to exit MS-LINK, and then reinvoke MS-LINK and start over. If the error has been typed, but not entered, you may delete the erroneous characters on that line only.

INVOKING MS-LINK

MS-LINK may be invoked in three ways. By the first method, you enter the commands in response to individual prompts. By the second method, you enter all commands on the line used to invoke MS-LINK. By the third method, you create a response file that contains all the necessary commands. The following table summarizes these methods. A description of each method follows the table.

Table 6-1. Summary of Methods Used to Invoke MS-LINK

Method:	Syntax:
Method 1	LINK
Method 2	LINK *filenames[/switches]*
Method 3	LINK *@filename*

Method 1: Using Command Prompts

To start MS-LINK by entering responses to all the command prompts (Method 1), first enter:

```
LINK
```

MS-LINK loads into memory. Four prompts are then displayed one at a time. Answer these prompts to direct MS-LINK to perform specific tasks.

One or more switches may be typed at the end of each line, preceded by a slash (see the description of switches in Tables 6-2 and 6-3). If switches are not selected, MS-LINK does not perform any of the switch functions described.

Table 6-2.
Command Prompt Summary

Prompt:	Responses:
Object Modules [.OBJ]:	List .OBJ files to be linked, separated by blank spaces or plus signs (+). If a plus sign is the last character entered, the prompt reappears. No default response is required.
Run File [object-file.EXE]:	Supply the *filename* for the executable object code. The default is first-object-filename.EXE.
List File [NUL.MAP]:	Supply the *filename* for the listing. The default is NUL.MAP
Libraries [.LIB]:	List filenames to be searched, separated by blank spaces or plus signs (+). If a plus sign is the last character entered, the prompt reappears. The default is no search.

Table 6-3.
MS-LINK Switch Parameters

Switch:	Action:
/DSALLOCATE	Loads data at the high end of the data segment. Required for Pascal and FORTRAN programs.
/HIGH	Places run file as high as possible in memory. Do not use with Pascal or FORTRAN programs.
/LINENUMBERS	Includes line numbers in the list file.
/MAP	Lists all global symbols with definitions.
/PAUSE	Halts the linker session and waits for the ENTER key to be pressed.
/STACK:number	Sets fixed stack size in the run file.

When the last prompt has been answered, MS-LINK begins linking automatically without further command. When the linking session is finished, MS-LINK exits to the operating system. If the linking session is unsuccessful, MS-LINK returns the appropriate error message.

Method 2: Command Line

To start MS-LINK using Method 2, enter all the commands on one line. The entries following MS-LINK are responses to the command prompts. The entry fields for the different prompts must be separated by commas. Use the syntax shown in the following example.

Enter:

```
LINK    object-list,runfile,listfile,lib-list[/switch...]
```

The entries following LINK are responses to the command prompts. The entry fields for the different prompts must be separated by commas.

object-list is a list of object modules, separated by plus signs.

runfile is the name of the file to receive the executable output.

listfile is the name of the file to receive the listing.

lib-list is a list of library modules to be searched.

/switch is any optional switch, which may be placed following any of the response entries (just before any of the commas or after *lib list,* as shown).

To select the default for a field, enter a second comma without a space between the two.

Example:

```
LINK  FUN+TEXT+TABLE+CARE/P/M,,FUNLIST,COB.LIB
```

This example causes MS-LINK to be loaded, then causes the object modules FUN.OBJ, TEXT.OBJ, TABLE.OBJ, and CARE.OBJ to be loaded. MS-LINK then pauses (caused by the /P switch). When you press any key, MS-LINK links the object modules, produces a global symbol map (the /M switch), defaults to FUN.EXE run file, creates a list file named FUNLIST.MAP, and searches the library file COB.LIB.

Method 3: Response File

To start MS-LINK using Method 3, enter:

```
LINK @[d:][pathname]filespec
```

where:

[*d*:] is the optional drive designator.

[*pathname*] is the optional drive and pathname specifier.

filespec, in this case, is the name of a response file. A response file contains answers to the MS-LINK prompts (shown under Method 1 for invoking), and may also contain any of the switches. Method 3 permits you to conduct the link session without interactive (direct) responses to the MS-LINK prompts.

NOTE: Before using Method 3 to invoke MS-LINK, you must first create the response file. A response file has text lines, one for each prompt. Responses and command prompts must appear in the same order.

Use switches and command characters in the response file the same as they are used for responses entered from the keyboard.

When the link session begins, each prompt is displayed in turn with the responses from the response file. The response file must contain valid responses for all the prompts, either in the form of filenames, or the semicolon command character. When MS-LINK prompts for a response not found in the response file, it waits for a keyboard entry. When a legal response has been entered, MS-LINK continues the link session.

Error messages and system messages produced during the linking process are described in Appendix A, MESSAGES.

CHAPTER 7

DEBUG

7-1. INTRODUCTION 7-1
7-2. STARTING DEBUG 7-2
7-3. DEBUG COMMANDS 7-3
DEBUG Commands Summary 7-4
Parameters 7-5
A (ASSEMBLE) Command 7-9
C (COMPARE) Command 7-12
D (DUMP) Command 7-13
E (ENTER) Command 7-16
F (FILL) Command 7-18
G (GO) Command 7-19
H (HEX) Command 7-21
I (INPUT) Command 7-22
L (LOAD) Command 7-23
M (MOVE) Command 7-25
N (NAME) Command 7-26
O (OUTPUT) Command 7-29
Q (QUIT) Command 7-30
R (REGISTER) Command 7-31
S (SEARCH) Command 7-34
T (TRACE) Command 7-35
U (UNASSEMBLE) Command 7-36
W (WRITE) Command 7-39

7-1. INTRODUCTION

The DEBUG program is used to provide a controlled testing environment for binary and executable object files. DEBUG eliminates the need to reassemble a program to see whether a problem has been fixed by a change. It permits the alteration of file or CPU register contents. After a file is altered, DEBUG can immediately reexecute a program to check the validity of the changes.

7-2. STARTING DEBUG

To start the DEBUG program, enter:

```
DEBUG  [d:] [pathname] [filespec] [arglist]
```

Segment register initialization is dependent upon the file to be loaded. The information to initialize the segment registers is in the load file. The BX:CX registers are loaded with the number of bytes placed in memory.

arglist may be specified if *filespec* is present. These file specification parameters and switches are passed to the program *filespec*. When *filespec* is loaded into memory, it is loaded as if it had been invoked with the following command:

filespec arglist

Here, *filespec* is the file to be debugged, and *arglist* is the rest of the command line.

If no *filespec* is specified, DEBUG is invoked as follows:

```
DEBUG
```

DEBUG returns the prompt (—), signaling that it is ready to accept your commands. Since no file specification has been given, current memory, disk sectors, or disk files can be worked on by invoking later commands.

For example, for *filespec*, you could enter:

```
DEBUG  COMP.COM
```

At start-up, the segment registers are set to the bottom of free memory, the instruction pointer is set to 0100 hex, the stack pointer is set to the end of the segment (or the bottom portion of COMMAND.COM), all flags are cleared, and the remaining registers are set to zero.

DEBUG loads COMP.COM into memory, starting at 100 hexadecimal in the lowest available segment. The BX:CX registers are loaded with the number of bytes placed in memory.

7-3. DEBUG COMMANDS

Each DEBUG command consists of a single letter followed by one or more parameters. Additionally, the control characters and the special editing function keys described in Chapter 5, EDLIN, all apply to DEBUG.

All DEBUG commands may be aborted at any time by pressing ^BREAK. ^NUM LOCK suspends the display, so that you can read it before it scrolls up out of sight. Pressing any key other than ^NUM LOCK restarts the display.

If a syntax error occurs in a DEBUG command, DEBUG displays the command line and indicates the error with a caret (^) and the word ERROR.

For example:

```
DCS:100 CS:110
          ^ERROR
```

All commands and parameters can be entered in upper and/or lower case.

The DEBUG commands are summarized below, and are described in detail, following the description of command parameters.

DEBUG COMMANDS SUMMARY

A list of the commands used in DEBUG follows:

Command:	**Syntax:**
A (ASSEMBLE)	A *[address]*
C (COMPARE)	C *range address*
D (DUMP)	D *[address* [L *value]]* or D *[range]*
E (ENTER)	E *address [list]*
F (FILL)	F *range list*
G (GO)	G *[=address] [address [address]]*
H (HEX)	H *value value*
I (INPUT)	I *address*
L (LOAD)	L *[address [d sector sector]]*
M (MOVE)	M *range address*
N (NAME)	N *filespec [filespec]*
O (OUTPUT)	O *address byte*
Q (QUIT)	Q
R (REGISTER)	R *[register name]*
S (SEARCH)	S *range list*

Command:	**Syntax:**
T (TRACE)	T [=*address*] [*value*]
U (UNASSEMBLE)	U [*address*] [L *value*] or U [*range*]
W (WRITE)	W [*address* [*d* *sector* *sector*]]

Parameters

As shown in the summary, all DEBUG commands accept parameters, except the QUIT command. Parameters can be separated by delimiters (spaces or commas), but a delimiter is required only between two consecutive hexadecimal values. The following commands are equivalent:

```
DCS:100 110
D   CS:100 110
D,CS:100,110
```

Parameter:	**Definition:**
d:	A 1-digit hexadecimal value indicating which drive a file is loaded from or written to. The valid values are 0, 1, and 2. These values designate the drives as follows: 0 = A:, 1 = B:, and 2 = C:.
byte	A 1- or 2-digit hexadecimal value to be placed in, or read from, an address or register.
sector sector	A 1- to 3-digit hexadecimal value used to indicate the logical sector number on the disk, and the number of disk sectors to be written or loaded. Numbering begins at track 0, sector 1, head 0. Sector size equals 512 bytes.

Table 7-1.
(Partial) Logical to Physical Mapping Table

	Type of Diskette											
	Single-Sided						Double-Sided					
	8 Sector			9 Sector			8 Sector			9 Sector		
Logical Sector	Track	Sector	Head	Track	Sector	Head	Track	Sector	Head	Track	Sector	Head
0	0	1	(0)	0	1	(0)	0	1	0	0	1	0
1	0	2	(0)	0	2	(0)	0	2	0	0	2	0
7	0	8	(0)	0	8	(0)	0	8	0	0	8	0
8	1	1	(0)	0	9	(0)	0	1	1	0	9	0
15	1	8	(0)	1	7	(0)	0	8	1	0	7	1
16	2	1	(0)	1	8	(0)	1	1	0	0	8	1
17	2	2	(0)	1	9	(0)	1	2	0	0	9	1
18	2	3	(0)	2	1	(0)	1	3	0	1	1	0

Parameter:	**Definition:**
value	One to four hexadecimal digits used to specify: • The numbers you want to add or subtract (see the HEX command). • The number of bytes to be operated on, or the number of instructions you want the TRACE command to execute.
address	A one- or two-part designation consisting of either an alphabetic segment register designation plus an offset value, or a 4-digit segment address plus an offset value, or a 16-bit port address, or an offset value alone. If the segment designation or segment address is omitted, the default segment is used. DS is the default seg-

Parameter: **Definition:**

ment for all commands except GO, LOAD, TRACE, UNASSEMBLE, and WRITE, for which the default segment is CS. All numeric values are hexadecimal. For example:

```
CS:0100
04BA:0100
```

The colon is required between a segment designation (whether numeric or alphabetic) and an offset.

range Enter either two addresses (*address address*), or one address, an L, and a value (where *value* is the number of bytes where the command operates).

address address
or
address L *value*

Example:

```
CS:100   110
or
CS:100   L   10
```

The following is illegal, and results in an error message:

```
CS:100   CS:110
           ^ERROR
```

The limit for *range* is 10000 hex.

Parameter:	**Definition:**
list	A series of byte values or strings. *list* must be the last parameter on the command line. For example: FCS:100 L 100 42 45 52 54 41 In this example, *list* is the last five bytes in the command line. See the FILL command for a detailed explanation of this example.
string	Any number of characters enclosed in quotation marks. Quotation marks may be either single (') or double ("). Within a string, the quotation marks that were not used to delimit the string can be used freely as literals. If the delimiter quotation marks must appear within a string, they must be doubled. For example, the following strings are legal: ‘This “string” is valid.’ ‘This ‘ ‘string’ ’ is valid.’ ‘This ‘string’ is not.’ “This ‘string’ is valid.” “This “ “string” ” is valid.” “This “string” is not.” Double quotation marks are not necessary in the following strings: “This ‘ ‘string’ ’ is not necessary.” ‘This “ “string” ” is not necessary.’ The ASCII values of the characters in the string are used as a list of byte values.

A (ASSEMBLE) Command

Purpose: To assemble 8086/8087/8088 mnemonics directly into memory.

Syntax: A [*address*]

Comments: If a syntax error is encountered, DEBUG responds with the ^ERROR message and redisplays the current assembly address.

All numeric values are hexadecimal and can be entered as one to four characters.

Prefix mnemonics must be entered in front of the opcode to which they refer. They can also be entered on a separate line.

The segment override mnemonics are CS:, DS:, ES:, and SS:.

String manipulation mnemonics must state word or byte. For example, MOVSW must be used to move word strings, and MOVSB must be used to move byte strings.

The mnemonic for the far return is RETF.

The assembler automatically assembles short, near, or far jumps and calls, depending on byte displacement to the destination address. These may be overridden with the NEAR or FAR prefix. For example:

```
0B00:0500  JMP  502 ; a 2-byte short jump
0B00:0502  JMP  NEAR  505 ; a 3-byte near jump
0B00:0505  JMP  FAR  50A ; a 5-byte far jump
```

The NEAR prefix can be abbreviated to NE, but the FAR prefix cannot be abbreviated.

DEBUG cannot tell whether some operands refer to a word memory location or a byte memory location. In this case, the data type must be stated with the prefix WORD PTR or BYTE PTR. DEBUG also accepts the abbreviations WO and BY for these prefixes. For example:

```
NEG   BYTE   PTR  [128]
DEC   WO   [SI]
```

DEBUG cannot tell whether an operand refers to a memory location or to an immediate operand. Operands enclosed in square brackets are assumed to refer to memory. For example:

```
MOV   AX,21     Load AX with 21H
MOV   AX,[21]   Load AX with the contents of
                memory location 21H
```

DEBUG includes two pseudo-instructions. The DB opcode assembles byte values directly into memory. The DW opcode assembles word values directly into memory. For example:

```
DB   1,2,3,4,“THIS IS AN EXAMPLE”
DB   ‘THIS IS A QUOTE:” ’
DB   “THIS IS A QUOTE:’ ”

DW   1000,2000,3000,“BACH”
```

All forms of the register indirect commands are supported. For example:

```
ADD     BX,34[BP+2].[SI-1]
POP     [BP+DI]
PUSH    [SI]
```

All opcode synonyms are supported. For example:

```
LOOPZ   100
LOOPE   100

JA      200
JNBE    200
```

For 8087 opcodes, the WAIT or FWAIT prefix must be specified. For example:

FWAIT FADD ST,ST(3)	This line assembles an FWAIT prefix
FLD TBYTE PTR [BX]	This line does not

C (COMPARE) Command

Purpose: To compare a portion of memory specified by a range to a portion of the same size beginning at a specified address.

Syntax: C *range address*

Comments: If the two areas of memory are identical, there is no display, and DEBUG returns with the MS-DOS prompt. If there are differences, they are displayed as:

address1 byte1 byte2 address2

Example:

The following commands have the same effect:

```
C   100,1FF   300
C   100L100   300
```

Each command compares the block of memory from 100 to 1FFH with the block of memory from 300 to 3FFH.

D (DUMP) Command

Purpose: To display the memory contents of either a single address, a range of addresses, or the number of bytes specified by a value, beginning at the address specified.

Syntax: D [*address* [L *value*]]
or
D [*range*]

Comments: If only a single address is specified, the contents of 128 bytes are displayed.

If a range of addresses is specified, the contents of the range are displayed.

If the D command is entered without parameters, the result is the same as if you specified a single address, except that the display begins at the current location in the data segment (DS).

The dump is displayed in two portions: a hexadecimal dump (each byte is shown in hexadecimal value) and an ASCII dump (the bytes are shown in ASCII characters). Nonprinting characters are denoted by a period (.) in the ASCII portion of the display.

Each displayed line begins on a 16-byte boundary, and the first and last line may contain fewer bytes than the rest of the displayed lines.

NOTE: If the display is only 40 characters wide, DUMP displays 8 bytes at a time, and the ASCII values wrap to the next line. If, however, the display is 80 columns wide, only 16 bytes of data will be displayed per line, and the values do not wrap to the next line. For example:

One line for an 80-column display:

```
04B5:0100  50  48  4F  54  4F  47  52  41
   —  50  48  49  43  20  53  45  52
   PHOTOGRAPHIC SER
```

Two lines, broken as follows, for a 40-column display:

```
04B5:0100 50 48 4F 54 4F 47 52 41   PHO
TOGRA

04B5:0108 50 48 49 43 20 53 45 52
        PHIC SER
```

When you enter:

```
D
```

the display is formatted as previously described. Each line of the display begins with an address incremented by 16 bytes from the address on the previous line. Each subsequent D (entered without parameters) displays the bytes immediately following those last displayed.

If you enter the command:

```
DSC:100   L20
```

the display is formatted as described above, with all bytes within the range of 100H to 1FFH in the CS segment displayed.

If you enter the command:

```
DSC:100   115
```

the display is formatted as described above, with all the bytes in the range from 100H to 115H in the CS segment displayed.

E (ENTER) Command

Purpose: To display an address and its contents, and wait for input.

Syntax: E *address* [*list*]

Comments: If the optional list of hexadecimal values is entered, the replacement of byte values occurs automatically. If an error occurs, no byte values are changed.

If *address* is entered without the optional *list*, DEBUG displays the address and its contents on the next line and waits for your input. Use the ENTER command in one of the following ways:

1) Replace a byte value with a value you enter. Enter the new value after the current value. If the new value is not a legal hexadecimal value, or if more than two digits are entered, the illegal or extra character is not entered.

2) Press the SPACEBAR to advance to the next byte. To change the value, enter the new value as described above. If you space beyond an 8-byte boundary, DEBUG starts a new display line with the address displayed at the beginning of the line.

3) Enter a hyphen (-) to move to the byte position before the current one. When the hyphen is entered, a new line is started with the address and its byte value displayed.

4) Press the ENTER key to terminate the ENTER command. The ENTER key may be pressed at any byte position.

Examples:

```
ECS:100
```

DEBUG displays:

```
04BA:0100   EB.__
```

To change this value to 41, enter 41 as shown:

```
04BA:0100   EB.41__
```

To step through the subsequent bytes, press the SPACEBAR:

```
04BA:0100   EB.41    10.    00.    BC.__
```

To change BC to 42:

```
04BA:0100   EB.41    10.    00.    BC.42__
```

To correct 10 to 6F, enter the hyphen as many times as needed to return to byte 0101 (value 10), and then replace 10 with 6F:

```
04BA:0100   EB.41    10.    00.    BC.42-

04BA:0102   00.-
04BA:0101   10.6F__
```

When you press the ENTER key, it ends the ENTER command, and returns to the DEBUG command level.

F (FILL) Command

Purpose: To fill the addresses in a range with the values in the specified list.

Syntax: F *range list*

Comments: If the specified range contains more bytes than the number of values in the specified list, the list is used repeatedly until all bytes in the range are filled.

If the list contains more values than the number of bytes in the range, the extra values in the list are ignored.

If any of the memory in the range is not valid (bad or nonexistent), the error is propagated into all succeeding locations.

The FILL command does not abort as the ENTER command does. The FILL command is a multiple version of the ENTER command that allows you to change more than one address at a time.

Example:

```
F04BA:100  L  100  42  45  52  54  41
```

DEBUG fills memory locations 04BA:100 through 04BA:1FF with the bytes specified. The five values are repeated until all 100H bytes are filled.

G (GO) Command

Purpose: To execute the program currently in memory.

Syntax: G [=*address*] [*address* [*address*]]

Comments: If the GO command is entered alone, the program executes as if it had run outside DEBUG.

If =*address* is set, execution begins at the address specified. If the segment designation is omitted from =*address*, only the instruction pointer is set. If the segment designation is included in =*address*, both the CS segment and the instruction pointer are set. The equal sign (=) is required so that DEBUG can distinguish the start address from the breakpoint addresses.

When program execution reaches the address of a breakpoint, execution halts and control is returned to DEBUG. The next instruction to be executed is decoded. Then all registers, flags, and the decoded next instruction are displayed for the last instruction executed. The result is the same as if you had entered the REGISTER command for the breakpoint address. Enter the GO command again to resume execution.

As many as 10 breakpoints can be set. Breakpoints can be set only at addresses containing the first byte of an 8086 opcode. If more than 10 breakpoints are set, DEBUG returns the BP error message.

The user stack pointer must be valid, and must have six bytes available for this command.

The GO command uses an IRET instruction to cause a jump to the program under test. The user stack pointer is set, and the user flags, code segment register, and instruction pointer are pushed onto the user stack. If the user stack is not valid, or is too small, MS-DOS may crash.

An interrupt code (0CCH) is placed at the specified breakpoint addresses. When an instruction with the breakpoint code is encountered, all breakpoint addresses are restored to their original instructions. If execution is not halted at one of the breakpoints, the interrupt codes are not replaced with the original instructions.

Example:

```
GCS:7550
```

The program currently in memory executes up to the address 7550 in the CS segment. DEBUG displays registers and flags, and the GO command is terminated.

After a breakpoint has been encountered, if the GO command is entered again, and the program executes as if you had entered the filename at the MS-DOS command level. The only difference is that program execution begins at the instruction after the breakpoint, rather than at the usual start address.

H (HEX) Command

Purpose: To perform hexadecimal arithmetic on two parameters.

Syntax: H *value value*

Comments: DEBUG adds the two parameters, then subtracts the second parameter from the first. The results of the arithmetic are displayed on one line; first the sum, then the difference.

Examples:

```
H19F   10A
```

DEBUG performs the calculations and returns the result:

```
02A9   0095
```

I (INPUT) Command

Purpose: To input and display one byte from the port specified by an address.

Syntax: I *address*

Comments: A 16-bit port address is allowed.

Example:

```
I2F8
```

If the byte at the port is 42H, DEBUG inputs the byte and displays the value:

```
42
```

L (LOAD) Command

Purpose: To load a file into memory.

Syntax: L [*address* [*d sector sector*]]

Comments: The file must have been named either with the DEBUG invocation command or with the NAME command. Either the invocation or the NAME command formats a filename properly in the normal format of a file control block at CS:5C.

If the LOAD command is given without any parameters, DEBUG loads the file into memory, beginning at address CS:100, and sets BX:CX to the number of bytes loaded.

Example:

```
A >DEBUG
-NFILE.COM
```

To load FILE.COM, enter:

```
L
```

If the LOAD command is entered with an address parameter, loading begins at the memory address specified.

If LOAD is entered with all parameters, absolute disk sectors are loaded, rather than a file. The sectors are taken from the drive specified. The drive specifier is numeric; for example, 0 = A: or 2 = C:.

DEBUG begins loading with the first sector specified, and continues until the number of sectors specified in the second sector have been loaded.

Example:

```
L04BA:100   1   0F   6D
```

DEBUG loads 109 (6D hex) sectors from Drive B:, beginning with logical sector number 15 (0F hex), into memory, beginning at address 04BA:0100. When the sectors have been loaded, the DEBUG prompt returns.

If the file has an .EXE extension, it is relocated to the load address specified in the header of the .EXE file; the address parameter is always ignored for .EXE files. The header is stripped off the .EXE file before it is loaded into memory. The size of an .EXE file on disk, therefore, differs from its size in memory.

If the file named by the NAME command, or specified on invocation, is a .HEX file, entering LOAD with no parameters causes loading of the file to begin at the address specified in the .HEX file.

If the LOAD command includes the optional address, DEBUG adds the address specified in LOAD to the address found in the .HEX file to determine the start address for loading the file.

M (MOVE) Command

Purpose: To move the block of memory specified by a range to the location beginning at the specified address.

Syntax: M *range address*

Comments: Overlapping moves (moves where part of the block overlaps some of the current addresses) are always performed without loss of data. Addresses that could be overwritten are moved first.

When moving from higher addresses to lower addresses, data is moved beginning at the block's lowest address and working towards the highest. When moving from lower addresses to higher addresses, data is moved beginning at the block's highest address and working towards the lowest.

If the addresses in the block being moved will not have new data written to them, the existing data remains. The sequence of the move is important because the MOVE command copies the data from one area into another, in the sequence described, and writes over the new addresses.

Example:

Enter:

```
MCS:100   110   CS:500
```

DEBUG first moves data from address CS:110 to address CS:510; then CS:10F to CS:50F, and so on, until CS:100 is moved to CS:500. Enter the DUMP command, using the address entered for the MOVE command, to review the results of the move.

N (NAME) Command

Purpose: To set file specifications.

Syntax: N *filespec* [*filespec*]

Comments: The NAME command performs two distinct functions, both associated with file specifications.

First, NAME is used to assign a file specification for a later LOAD or WRITE command. If you invoke DEBUG without naming any file to be debugged, the NAME command must be given before a file can be loaded.

Second, NAME is used to assign file specification parameters to the file being debugged. In this case, NAME accepts a list of parameters used by the file being debugged.

These functions overlap, as in the following set of DEBUG commands:

```
-NFILE.EXE
-L
-G
```

NAME assigns the name FILE.EXE to the file specification to be used in any later LOAD or WRITE commands.

NAME also assigns the name FILE.EXE to the first filespec parameter to be used by any program that is later debugged.

LOAD loads FILE.EXE into memory.

GO causes FILE.EXE to be executed with FILE.EXE as the single filespec parameter. For example, FILE.EXE is executed as if it had been typed at the command level.

Example:

```
-NFILE1.EXE
-L
-NFILE2.DAT  FILE3.DAT
-G
```

NAME sets FILE1.EXE as the filespec for the subsequent LOAD command. The LOAD command loads FILE1.EXE into memory, and then the NAME command is used again, to specify the parameters to be used by FILE1.EXE.

When the GO command is executed, FILE1.EXE is executed as if FILE1 FILE2.DAT FILE3.DAT had been entered at the MS-DOS command level.

> If a WRITE command is executed at this point, FILE1.EXE (the file being debugged) is saved with the name FILE2.DAT. To avoid this result, always execute a NAME command before either a LOAD or WRITE command.

Four regions of memory can be affected by the NAME command:

CS:5C	FCB for file 1.
CS:6C	FCB for file 2.
CS:80	Count of characters.
CS:81	All characters entered.

A file control block (FCB) for the first filespec parameter given to the NAME command is created at CS:5C. If a second filespec parameter is given, an FCB is created for it beginning at CS:6C. The number of characters entered in the NAME command, exclusive of N, is given at location CS:80. This actual stream of characters given by the NAME command, exclusive of N, begins at CS:81. The stream of characters may contain any switches and delimiters that would be legal in any command typed at the MS-DOS command level.

Example:

```
DEBUG  PROG.COM
-NPARAM1  PARAM2/C
-G
-
```

The GO command executes the file in memory as if the following command line had been entered:

```
PROG  PARAM1  PARAM2/C
```

Testing and debugging reflect a normal runtime environment for PROG.COM.

O (OUTPUT) Command

Purpose: To send the specified byte to the output port specified by an address.

Syntax: O *address byte*

Comments: A 16-bit port address is allowed.

Example:

Enter:

```
O2F8 4F
```

DEBUG outputs the byte value 4F to output port 2F8.

Q (QUIT) Command

Purpose: To terminate DEBUG.

Syntax: Q

Comments: The QUIT command exits DEBUG without saving the file currently being operated on, and returns to the MS-DOS command level.

QUIT takes no parameters.

Example:

Enter:

```
Q
```

DEBUG is terminated, and control returns to the MS-DOS command level.

R (REGISTER) Command

Purpose: To display the contents of one or more CPU registers.

Syntax: R [*register name*]

Comments: If no register name is entered, the REGISTER command dumps the register-save area, and displays the contents of all registers and flags.

If a register name is entered, the 16-bit value of that register is displayed in hexadecimal, and a colon appears as a prompt. Either enter a value to change the register, or press the ENTER key if no change is desired.

The only valid register names are:

AX
BX
CX
DX
SP
BP
SI
DI
DS
ES
SS
CS
IP
PC
F

IP and PC both refer to the instruction pointer.

Any other entry for register name results in a BR error message.

If F is entered as the register name, DEBUG displays a two-character alphabetic code. To alter the flag, enter the opposite two-letter code. The flags are either set or cleared. The flags and their set and clear codes are listed in Table 7-2.

Table 7-2.
Set and Clear Codes for DEBUG Flags

Flag Name:	Set:	Clear:
Overflow	OV	NV
Direction	DN (decrement)	UP (increment)
Interrupt	EI (enabled)	DI (disabled)
Sign	NG (negative)	PL (positive)
Zero	ZR	NZ
Auxiliary Carry	AC	NA
Parity	PE (even)	PO (odd)
Carry	CY	NC

When you enter RF, the flags are displayed in the order shown above. At the end of the list of flags, DEBUG displays a static hyphen (-).

New flag values may be entered as alphabetic pairs in any order. Spaces are not required between the flag entries.

To exit the REGISTER command, press the ENTER key. Flags that do not have new values entered remain unchanged.

If more than one value is entered for a flag, DEBUG returns a DF error message. If a flag code other than those shown above is entered, DEBUG returns a BF error message. In either case, flags before the error are changed, and flags after the error are not changed.

Example:

When you enter R, DEBUG displays all registers, flags, and the decoded instruction for the current location.

When you enter RF, DEBUG displays the flags:

```
NV UP DI NG NZ AC PE NC -__
```

Enter any valid flag designations, in any order, with or without spaces. For example, enter:

```
NV UP DI NG NZ AC PE NC - PLEICY
```

DEBUG responds with only the DEBUG prompt. To see the changes, enter R or RF:

```
RF
NV UP EI PL NZ AC PE CY - __
```

S (SEARCH) Command

Purpose: To search the specified range for the specified list of bytes.

Syntax: S *range list*

Comments: The list may contain one or more bytes, each separated by a space or comma. The list may also contain a string.

If the list contains more than one byte, only the first address of the byte string is returned.

If the list contains only one byte, all addresses of the byte in the range are displayed.

Example:

Enter:

```
SCS:100   110   41
```

DEBUG displays:

```
04BA:0104
04BA:010D
```

If there are not matches, the prompt (-) is returned with no messages.

T (TRACE) Command

Purpose: To execute one or more instructions, and display the contents of all registers, flags, and the decoded instruction.

Syntax: T [=*address*] [*value*]

Comments: If the optional =*address* is entered, tracing occurs at the address specified. If =*address* includes the segment designation, then both CS and the instruction pointer are specified. If =*address* omits the segment designation, only the instruction pointer is specified. The optional *value* causes DEBUG to execute and trace the number of steps specified by *value*.

Example:

When you enter T, DEBUG returns a display of the registers, flags, and decoded instructions for that one instruction.

Enter:

```
T=011A  10
```

DEBUG executes 16 (10 hex) instructions, beginning at 011A in the current segment, and then displays all registers and flags for each instruction as it is executed. The display scrolls up until the last instruction is executed, then stops and displays the register and flag values for the last few instructions performed. ^NUM LOCK suspends the display at any point, so that the registers and flags can be studied for any instruction.

U (UNASSEMBLE) Command

Purpose: To unassemble bytes and display the source statements that correspond to them, along with addresses and byte values.

Syntax:

U [*address*] [L *value*]
or
U [*range*]

Comments: The display of unassembled code looks like a listing for an assembled file.

If the UNASSEMBLE command is entered without parameters, 20 hexadecimal bytes are unassembled to show corresponding instructions.

If the UNASSEMBLE command is entered with the range parameter, DEBUG unassembles all bytes in the range.

If there are fewer bytes in the range than the number displayed when U is entered without parameters, only the bytes in the range are displayed.

If the UNASSEMBLE command is entered with the address parameter, DEBUG unassembles the default number of bytes, beginning at the address specified.

If the UNASSEMBLE command is entered with the address and L value parameters, DEBUG unassembles all bytes, beginning at the address specified, for the number of lines specified by the L value parameter.

Entering U with the address and L value parameters or with a range overrides the default limit (20H bytes).

Example:

Enter:

```
U04BA:100
```

DEBUG unassembles 16 bytes, beginning at address 04BA:0100:

```
04BA:0100  206472  AND   [SI+72],AH
04BA:0103  69      DB    69
04BA:0104  7665    JBE   016B
04BA:0106  207370  AND   [BP+DI+70],DH
04BA:0109  65      DB    65
04BA:010A  63      DB    63
04BA:010B  69      DB    69
04BA:010C  66      DB    66
04BA:010D  69      DB    69
04BA:010E  63      DB    63
04BA:010F  61      DB    61
```

Enter:

```
U04BA:0100  0108
```

The display shows:

```
04BA:0100   206472   AND   [SI + 72],AH
04BA:0103   69       DB    69
04BA:0104   7665     JBE   016B
04BA:0106   207370   AND   [BP + DI + 70],DH
```

Enter:

```
U04BA:100   120
```

The display appears exactly as previously shown with the UCS:100 command.

Enter:

```
UCS:100   L   20
```

All the bytes from address CS:100 through address CS:120 are unassembled and displayed. This applies to both display formats.

If the bytes in some addresses are altered, the instruction statements are altered. The UNASSEMBLE command can be entered for the changed locations, the new instructions viewed, and the unassembled code used to edit the source file.

W (WRITE) Command

Purpose: To write the file being debugged to a disk file.

Syntax: W [*address* [*d sector sector*]]

Comments: If W is entered without parameters, BX:CX must already be set to the number of bytes to be written. The file is written beginning from CS:100.

If the WRITE command is given with only an address, the file is written beginning at that address.

If a GO or TRACE command is given, BX:CX must be reset before using the WRITE command without parameters.

If a file is loaded and modified, the name, length, and starting address are all set correctly to save the modified file, as long as the length has not changed.

The file must have been named, either during DEBUG invocation, or with the NAME command. Both the invocation and the NAME command format a filename properly in the normal format of a file control block at CS:5C.

DEBUG writes the BX:CX number of bytes to the disk file. The debugged file is written to the disk from which it was loaded, and is written over the original file.

If the WRITE command is given with parameters, the write begins from the memory address specified, and the file is written to the drive specified. The drive specifier is numeric; 0 = A:, 1 = B:, or 2 = C:. DEBUG writes the file beginning at the logical sector number specified by the first sector, and continues until the number of sectors specified in the second sector have been written.

NOTE: Writing to absolute sectors is not recommended. Doing so bypasses the file handler.

Example:

Enter:

```
W
```

DEBUG writes out the file to a disk, and displays the DEBUG prompt:

```
W
-_
```

Enter:

```
WCS:100  1  37  2B
```

DEBUG writes out the contents of memory, beginning with the address CS:100, to the disk in Drive B:. The data written out starts in disk logical record number 37H, and consists of 2BH sectors.

APPENDIX A

MESSAGES

MESSAGES

This is an alphabetic list of messages produced by MS-DOS, with a brief description of each. The program or command that produces the message is given at the end of each message's description.

About to generate .EXE file
Change disks <hit ENTER>

When you specify the /PAUSE switch in the MS-LINK utility, this message is displayed. Insert a runfile diskette in the appropriate drive and press ENTER. (MS-LINK)

Access denied

You cannot write to a read-only protected file. (DEBUG)

All cylinders on disk are allocated.

No cylinders are available for creation of a partition. (FDISK)

All 4 fixed disk partitions are allocated.

Only four partitions are allowed. A new partition cannot be created unless one of the existing partitions is removed. (FDISK)

All files canceled by operator

The /T switch was used to terminate printing of the queue. (PRINT)

All specified file(s) are contiguous

All the files that you entered are written sequentially. (CHKDSK)

Allocation error for file, size adjusted

File was truncated at the end of the last valid sector because an invalid sector number was found in the file allocation table. (CHKDSK)

Ambiguous switch: z
An MS-LINK switch has not been clearly identified by the characters represented by z. Use more or sequential characters from the switch name. (MS-LINK)

Amount read less than size in header
The program in the named file was smaller than the header indicated it should be. Recompile, reassemble, and relink the program. (EXE2BIN)

An internal failure has occurred
Error from the MS-LINK program. Contact an Authorized COMPAQ Computer Dealer and describe the conditions that produced this message. (MS-LINK)

Attempt to access data outside of segment bounds, possibly bad object module
Probably a bad object file. (MS-LINK)

Attempted write-protect violation
You attempted to format a write-protected diskette. Insert another diskette, wait for the prompt, and press any key to start. (FORMAT)

Backup file sequence error
When files were backed up from the fixed disk, a file was recorded on more than one diskette. Now you are trying to restore the file with the diskettes in the wrong order. Restart RESTORE, inserting the diskettes in the proper sequence. (RESTORE)

Bad command or file name
An invalid internal command was entered. The file called *command-name*. COM or *command-name*.EXE cannot be found on the specified drive. (MS-DOS)

Bad numeric parameter
Numeric value for /STACK is not in digits. (MS-LINK)

Bad or missing Command Interpreter
Either a copy of COMMAND.COM was not found on the disk in the default drive, or an error occurred when the disk was loaded. Try a system reset. If the problem is not corrected, copy COMMAND.COM from a back-up diskette onto the disk reporting the error. (MS-DOS)

Bad or missing <filename>
MS-DOS cannot find the device driver specified by DEVICE = in ANSI.SYS, or some other type of error may have occurred while DOS was loading the driver. (MS-DOS)

Batch file missing
The batch file currently being executed was probably renamed or erased by MS-DOS. (MS-DOS)

Beginning cylinder access installed or invalid cylinders.
The specified starting cylinder is in an already-installed partition, or is an illegal cylinder.

BF
Bad flag. Retry the REGISTER command (R) with the correct flag code setting. (DEBUG)

BP
Breakpoints. The GO command cannot have more than ten breakpoints. (DEBUG)

BR
Bad register. Retry the REGISTER (R) command with a correct register name. (DEBUG)

Cannot do binary reads from a device
You attempted to copy using a /B switch with a device name. Omit the /B, or use /A instead. (COPY)

Cannot edit .BAK file—rename file
You attempted to edit a file with an extension of .BAK. .BAK files cannot be edited because this extension is reserved for

back-up copies. If you need this file for editing purposes, you must either rename the file with a different extension, or copy the .BAK file and give it a different extension. (EDLIN)

Cannot find file *object file*
Change diskette <hit ENTER>
Specified object file cannot be found on the disk. Insert the diskette with the specified object file and press ENTER. (MS-LINK)

Cannot find library file *library file*
Enter new drive letter:
Specified library was not found on the disk. Type the letter for the drive where the library is located. (MS-LINK)

Cannot load COMMAND, system halted
COMMAND.COM cannot be found as specified by COMSPEC, or the tracking of available memory by MS-DOS has been lost. Reload MS-DOS. (MS-DOS)

Cannot nest response file
Automatic response file contained @*filespec*. (MS-LINK)

Cannot open list file
There is not enough space left in the directory or on the disk. (MS-LINK)

Cannot open overlay
There is not enough space left in the directory or on the disk. (MS-LINK)

Cannot open response file
MS-LINK cannot find the automatic response file. (MS-LINK)

Cannot open temporary file
There is not enough space left in the directory or on the disk. (MS-LINK)

Cannot start COMMAND, exiting
Not enough memory is available to load a second copy of

COMMAND.COM, or the value specified by FILES= is too small. (MS-DOS)

COM*n*: *bbbb,p,d,s,t* initialized
The MODE command (asynchronous communications adapter command) has been used to specify the communications parameters. See the description of the MODE command. (MODE)

Compare error at offset xxxxxxxx
File 1 = xx
File 2 = xx
There are different values at the displayed hexadecimal offset in the files being compared. (COMP)

Compare error(s) on
Track xx, side xx
The disks being compared contain differing information at the track and side locations given. (DISKCOMP)

Compare more diskettes (Y/N)?
Enter Y to compare more diskettes; enter N to stop. (DISKCOMP)

Compare more files?
Enter Y to compare two more files; enter N to stop. (COMP)

Comparing x sectors per track, *n* side(s)
x indicates the number of sectors per track found on the first diskette. *n* indicates the number of sides that can be read from the first diskette. (DISKCOMP)

Contains invalid cluster,
file truncated
There is an invalid data area pointer in the named file. If the /F switch was specified, the file is truncated at the last good block of data. (CHKDSK)

Contains xxx non-contiguous blocks
The named file was not sequentially written; it was written in xxx blocks. The checking process will take longer. (CHKDSK)

Convert directory to file (Y/N)?
The named directory is too fragmented to be used as a directory any longer. Respond with Y to examine the file using DEBUG. (CHKDSK)

Convert lost chains to files (Y/N)?
Respond with Y to try to recover data in lost blocks. If you respond with N, the blocks are allocated to new files. (CHKDSK)

Copy another (Y/N)?
To copy another diskette, enter Y. To stop, enter N. (DISKCOPY)

Copy complete
All data on the source diskette has been copied to the target diskette. (DISKCOPY)

Copying x sectors per track, *n* side(s)
x indicates the number of sectors per track compared on the first diskette. *n* indicates the number of sides read from the first diskette. (DISKCOPY)

Data error writing drive X
Abort, Retry, Ignore?
See "Disk error reading drive *X*". (MS-DOS)

DF
Double flag. A single flag had conflicting codes specified. A flag is permitted to be changed only once per REGISTER (R) command. (DEBUG)

Disk boot failure
An error prevented MS-DOS from being loaded into memory. If the error persists, use a back-up MS-DOS disk. (MS-DOS)

Disk error reading (or writing) drive X
Abort, Retry, Ignore?

An error has occurred in a disk read or write operation. MS-DOS has retried the operation three times unsuccessfully.

When this message occurs, **do not** change disks.
(This message can be caused by using a double-sided diskette in a single-sided drive.)

MS-DOS waits for you to enter one of the following responses:

(A)bort. Terminate the program requesting the disk read or write.

(R)etry. Repeat the operation. This response is used when you have corrected the error (such as with a Not Ready or a Write Protect error).

(I)gnore. The system disregards the error.

Error recovery is usually attempted by entering responses in the following order:

R Retry the operation (the error may not recur).
A Terminate the program and try a new disk.
I Ignore the error condition (this action can cause data to be lost).

If DEBUG is being executed, the "Abort, Retry, Ignore" prompt does not appear. Press F3 to redisplay the LOAD or WRITE command, then press RETURN. (MS-DOS)

Disk error reading FAT

Unsuccessful attempt by CHKDSK to update the file allocation table. If the message is displayed twice, the disk cannot be used. (CHKDSK)

Disk error writing FAT X
Unsuccessful attempt by CHKDSK to update the file allocation table. If X = 1, the first FAT was not written. If X = 2, the second FAT was not written. If the message is displayed twice, the disk cannot be used. (CHKDSK)

Disk full — write not completed
An END edit command cannot be completed. The disk does not have enough free space to save the file. Some of the file may be saved on disk, but the remainder is lost. (EDLIN)

Disk not compatible
The specified disk cannot be formatted using the FORMAT command. (FORMAT)

Disk unsuitable for system disk
The disk has a bad track where MS-DOS should be. The disk can probably be used if you don't try to put system files on it. (FORMAT)

Diskette error
Replace and strike any key when ready
A disk read error occurred during system start-up. Replace with an MS-DOS diskette and press a key. (MS-DOS)

Diskette is not a backup diskette
The diskette was not created by BACKUP. Use the correct diskette. (RESTORE and BACKUP)

Divide overflow
There is an internal logic error, or a program gave an instruction to divide a number by zero. This has the same effect as ^BREAK. (MS-DOS)

Do you see the leftmost 9? (Y/N)
Prompt displayed when the MODE test pattern (T) has been displayed shifted to the right. The prompt is repeated until you respond Y. (MODE)

Do you see the rightmost 9? (Y/N)
Prompt displayed when the MODE test pattern (T) has been displayed shifted to the left. The prompt is repeated until you respond Y. (MODE)

Do you wish to use the entire fixed disk for MS-DOS (Y/N)? [Y]
Respond with Y to use the entire fixed disk for MS-DOS. Respond with N if MS-DOS is to be one of several installed operating systems. (FDISK)

Dup record too complex
An assembly language source module DUP record is too complex; 1024 bytes are required for expansion of a single DUP. Debug the machine language processor source and rerun MS-LINK. (MS-LINK)

Duplicate filename or file not found
The new filename you entered is already being used on this disk, or the file you tried to rename cannot be located on the disk. (RENAME)

Enter beginning cylinder: [xxxx]
Respond with the number of the cylinder where the MS-DOS partition will start. (FDISK)

Enter partition number to activate: [ ]
Respond with the number of the partition you want to be active. (FDISK)

Enter partition size: [xxxx]
Respond with a number in cylinders that indicates the size you want the MS-DOS partition to be. (FDISK)

Enter primary file name
Respond with the first of the two files you want compared. (COMP)

Enter 2nd file name or drive id

Respond with the second of the two files you want to compare, or with the drive specifier if the second file name is the same as the first. (COMP)

Entry Error

The last command entered contained a syntax error. Retype the command with the correct syntax and press ENTER. (EDLIN)

Entry has a bad attribute (or size or link)

If you specified the /F switch, CHKDSK tries to correct errors. One or two periods appearing in the message indicate which subdirectory contains the error. (CHKDSK)

EOF mark not found

An EOF was not found in the last block of the files being compared (usually when comparing nontext files). (COMP)

Error found, F parameter not specified
Corrections will not be written to disk

CHKDSK did not correct the errors it displays because you did not specify the /F switch. (CHKDSK)

Error in EXE file

The relocation information placed in the file by the MS-LINK program contains an error. (MS-DOS)

Error in EXE/HEX file

There are invalid records or characters in the file. (DEBUG)

Error loading operating system

The operating system cannot be loaded from your fixed disk. If the error continues, load the system from a diskette and reinstall MS-DOS on the fixed disk. (MS-DOS)

Error reading fixed disk

FDISK cannot read the fixed disk. Consult your Authorized COMPAQ Computer Dealer. (FDISK)

Error verifying fixed disk.

FDISK cannot verify the written data. Consult your Authorized COMPAQ Computer Dealer. (FDISK)

Error writing fixed disk

FDISK cannot write to the fixed disk. Consult your Authorized COMPAQ Computer Dealer. (FDISK)

Error writing to device

MS-DOS cannot write to the specified device. (MS-DOS)

Errors on list device indicate that it may be off-line. Please check.

You attempted to issue a PRINT command to an offline device. (PRINT)

EXE and HEX files cannot be written

DEBUG does not support the backwards conversion required by the data. (DEBUG)

EXEC failure

FILES= command specified too small a value, or MS-DOS tried to execute a command and encountered an error. (MS-DOS)

File allocation table bad, drive X
Abort, Retry, Ignore?

See "Disk error reading drive X". This message means that the copy in memory of one of the allocation tables has pointers to nonexistent blocks. The disk may be incorrectly formatted or unformatted. If the error persists, the disk may be unusable, or requires reformatting. (MS-DOS)

File canceled by operator

Printing has been terminated. (PRINT)

File cannot be converted

The .EXE file cannot be converted because it does not meet the necessary file criteria or has segment fixups. (EXE2BIN)

File cannot be copied onto itself

A copy of a file has been given the same name as the original file on the same disk as the original. (MS-DOS)

File creation error

A new filename cannot be added to the directory. Use CHKDSK to see if the directory is full or what other condition has caused the error. (MS-DOS)

File is cross-linked:
on cluster xx

Remedy the cross-linking by one of the following methods:

- deleting the original files
- copying both files
- editing the files

(CHKDSK)

File is currently being printed
File is in queue

Messages for information only. May appear individually or together. (PRINT)

File not found

A file specified in a command was not found. Enter a valid file specification and try again. (MS-DOS and commands)

Filespec* and *filespec

These are the names of the two files currently being compared. (COMP)

Files are different sizes

The files are not the same size and so cannot be compared; one contains data that the other does not. (COMP)

Files compare OK

The two files compared contain identical information. (COMP)

First cluster number is invalid, entry truncated

The file specified has an invalid data area pointer. If you specified the /F switch, the file is truncated to zero. (CHKDSK)

Fixed disk already has an MS-DOS partition

You cannot install a second MS-DOS partition on the fixed disk. Make another choice. (FDISK)

Fixup offset exceeds field width

An assembly language instruction refers to an address with a short instruction instead of a long instruction. Edit the assembly language source and reassemble. (MS-LINK)

Fixups needed — base segment (hex):

The .EXE file indicated that a load segment is needed. Enter the absolute segment address. (EXE2BIN)

FOR cannot be nested

There is more than one FOR command on a line. (BATCH)

Format failure

The target disk is unusable because of a disk error. (FORMAT)

Formatting while copying

The DISKCOPY utility is formatting unformatted tracks on the target diskette while copying data. (DISKCOPY)

Has invalid cluster, file truncated

An invalid cluster has caused part of the file to be lost. (CHKDSK)

Illegal device name

Printers must be specified as LPT1:, LPT2:, or LPT3:; asynchronous communications ports must be COM1: or COM2:. (MODE)

Incompatible diskette or drive types.

A read error has occurred while comparing diskettes. A diskette or drive may be single-sided, or defective. (DISKCOMP)

Incompatible drive types.

An error writing to the destination disk has occurred. The diskette or drive may be single-sided, or defective. (DISKCOPY)

Incompatible system size

The copy of MS-DOS on the target disk is smaller than the one being copied, so the system transfer does not occur. (SYS)

Incorrect MS-DOS version

You tried to enter a command that requires a different version of MS-DOS than the one you are using. (MS-DOS)

Input File Read Error

There is probably a bad object file. (MS-LINK)

Insert backup diskette xx in drive X
Strike any key when ready

Insert the new back-up diskette to continue restoring. (RESTORE)

Insert backup diskette xx in drive X
Warning! Diskette files will be erased
Strike any key when ready

Insert the back-up diskette and press a key to begin. (BACKUP)

Insert COMMAND.COM disk in drive X:
Strike any key when ready

COMMAND.COM cannot be found on the drive that originally loaded MS-DOS. (MS-DOS)

Insert disk to be recovered into
drive X: and press any key when ready

Insert the diskette that you want to recover into the drive and press a key to start. (RECOVER)

Insert disk with batch file
and strike any key when ready

The disk where the batch file is located was removed during processing. Batch processing cannot continue until the disk is inserted and a key is pressed. (MS-DOS)

Insert DOS disk in X:
and strike any key when ready

The indicated drive does not contain the MS-DOS diskette and, either MS-DOS is trying to load the command processor, or FORMAT or SYS is trying to load the MS-DOS files. (MS-DOS, FORMAT, and SYS)

Insert DOS diskette in drive A:
Press any key when ready...

The specified drive does not contain the MS-DOS diskette. (FORMAT and SYS)

Insert first diskette in drive X
Insert second diskette in drive X

Insert the first or second diskette in the indicated drive. When prompted, press a key to continue the comparison. (DISKCOMP)

Insert source diskette in drive X
Insert target diskette in drive X

Insert the proper diskette in the indicated drive and press a key when prompted. Copying will continue. (DISKCOPY)

Insufficient disk space

There is not enough free space on the disk for the file being written. Run CHKDSK to check the status of the disk. (MS-DOS and commands)

Insufficient memory

There is not enough memory to run a command. You must free some memory by writing files to a disk, by deleting a file, or by changing the value specified by BUFFERS =. (MS-DOS)

Insufficient room in root directory
Erase files from root and repeat CHKDSK

Data found by CHKDSK cannot be put into files because the directory is full. (CHKDSK)

Insufficient space on disk

A WRITE command was issued to a disk without enough free space for the data to be written. (DEBUG)

Intermediate file error during pipe

The root directory is full, pipe files cannot be located, or there is not enough space on the disk for the piped data. (MS-DOS)

Invalid baud rate specified

The allowed baud rates are 110, 150, 300, 600, 1200, 2400, 4800 or 9600. Enter the first two digits of the baud rate. (MODE)

Invalid characters in volume label

You entered a volume label that contained an illegal character. (FORMAT)

Invalid COMMAND.COM in drive X

The version of COMMAND.COM on the disk is incorrect. Replace it with the correct MS-DOS version when prompted. (MS-DOS)

Invalid current directory
Processing cannot continue

There is an invalid directory organization. Restart the system and rerun CHKDSK. If the error reoccurs, the diskette may be unusable. (CHKDSK)

Invalid date

You gave an invalid date or date delimiter. The only delimiters allowed are hyphens (-) or slashes (/). (MS-DOS)

Invalid device
You entered an improper device name. (CTTY)

Invalid directory
A pathname that you entered contains a nonexistent directory. (MS-DOS and commands)

Invalid drive in search path
You gave an incorrect drive designator in the PATH command. (MS-DOS)

Invalid drive specification
Invalid drive was specified in a command or parameter. (MS-DOS and commands)

Invalid format file
Library error. (MS-LINK)

Invalid number of parameters
You gave an incorrect number of parameters in a command. (Commands)

Invalid numeric parameter
A numeric parameter was not given in digits. (MS-LINK)

Invalid object module
An object module is incorrectly formed or incomplete (as when assembly is stopped in the middle). (MS-LINK)

Invalid parameter
Invalid switch specified. Retry with the proper switch. (MS-DOS and commands)

Invalid Parameter
No parameters were entered, or the first parameter is not 40, 80, BW40, BW80, CO40, CO80, or MONO. (MODE)

Invalid partition table
There is invalid partition information on the fixed disk. Use FDISK to correct the problem. (MS-DOS)

Invalid path
An error was detected in the directory structure. Try to locate it using CHKDSK. (TREE)

Invalid path, not directory or directory not empty
You either attempted to remove a directory that was not empty or you gave an invalid pathname. (RMDIR)

Invalid path or file name
You entered a nonexistent pathname or file specification. (COPY)

Invalid sub-directory
There is incorrect information in the named subdirectory. Try running CHKDSK using the /V switch. (CHKDSK)

Invalid switch: z
You entered an invalid linker parameter. (MS-LINK)

Invalid time
An illegal time was entered. Enter the time again. (TIME)

Label not found
A nonexistent label was specified with GOTO. (Batch)

Line too long
A REPLACE command was given a *string* that caused the line to be longer than the 254-character limit. EDLIN aborts the REPLACE command. Divide the line that is too long into shorter lines and retry the REPLACE command. (EDLIN)

List output is not assigned to device
You specified an invalid device for PRINT. (PRINT)

LPT#: not redirected
Setting the printer width or vertical spacing has canceled any previous redirection instructions. The parallel printer now receives its output instead of redirecting to a serial device. (MODE)

LPT#: redirected to COM*n*

Output designated for the indicated printer is being redirected to the serial device COM*n* (*n* = 1,2,3). (MODE)

LPT#: set for 80

Requesting standard type format has caused an attempt to set the printer line length to 80 characters. If the attempt is not successful, an error message follows. (MODE)

LPT#: set for 132

Requesting compressed type format has caused an attempt to set the printer line length to 132 characters. If the attempt is not successful, an error message follows. (MODE)

Maximum available partition in cylinders is xxxx beginning at cylinder xxxx.

This is the location of the largest amount of available space that can be used for an MS-DOS partition. (FDISK)

Memory allocation error
Cannot load COMMAND, system halted

The memory tracking area of MS-DOS has been lost. Reload MS-DOS. (MS-DOS)

Missing file name

Two file specifications are required; the second was not specified. (RENAME)

Missing operating system

MS-DOS could not be loaded from the fixed disk. Copy SYS to the fixed disk. (MS-DOS)

Must specify destination line number

A destination line number was not specified for a COPY or MOVE command. Reissue the command with a destination line number. (EDLIN)

Name of list device [PRN]:
Enter the name of the device that is to receive the print data. Just press ENTER to specify the first parallel printer (default choice). (PRINT)

No MS-DOS partition to remove.
There is no MS-DOS partition on your fixed disk to remove. Make another choice. (FDISK)

No fixed disk partitions to modify
You attempted to modify the active partition, and there are none to modify. Make another choice. (FDISK)

No free file handles
Cannot start COMMAND, exiting
There are too many files open for a second copy of COMMAND.COM to be loaded. Specify a larger number with FILES = and try. (MS-DOS)

No object modules specified
No object modules were named in the command line or in response to the prompt. MS-LINK needs the names of the files to be linked. (MS-LINK)

No path
MS-DOS cannot find a path to use to search for batch files and commands. (PATH)

No room for system on destination disk
The required reserved space for MS-DOS is not on the destination disk so the system cannot be transferred. If necessary, format a blank disk and copy the needed files. (SYS)

No room in directory for file
Because the directory is full, no editing changes can be recorded. (EDLIN)

No sub-directories exist
Only a root directory exists. (TREE)

Non-System disk or disk error
Replace and strike any key when ready

The directory does not contain the system files, or a disk read error occurred during system start-up. Insert an MS-DOS diskette and press a key. (MS-DOS)

Not enough room to merge the entire file

There is not enough room in memory to hold the file during a TRANSFER command; only part of the file was merged. Before you can transfer this file, you must free some memory by writing some files to a disk, or by deleting some files. (EDLIN)

Not found

The REPLACE *text* or SEARCH *text* commands could not locate the specified string. (EDLIN)

Not ready error reading drive X.
Correct, then strike any key

The disk drive door was open while a read was being attempted. (DISKCOMP and DISKCOPY)

Not ready error writing drive X.
Correct, then strike any key

The disk drive door was open while a write was being attempted. (DISKCOPY)

Out of environment space

There is not enough space in the environment storage area for the SET command to be processed. This condition usually occurs after you have loaded a program and then tried to add environment information. (MS-DOS)

Out of space on list file

This usually means there is not enough room on the disk for the list file. (MS-LINK)

Out of space on list file

This usually means there is not enough room on the disk for the run (.EXE) file. (MS-LINK)

Out of space on VM.TMP
No disk space remains to expand the VM.TMP file. (MS-LINK)

Parameters not compatible
You specified two incompatible switches (such as /V and /B). (FORMAT)

Parameters not compatible with fixed disk
You attempted to use the /1 or /8 switch with a fixed disk. (FORMAT)

Partition size cannot be zero.
A partition of size 0 was requested. (FDISK)

Present System Contains No Fixed Disks.
You specified the FDISK command and there is not a fixed disk to access. (FDISK)

Press any key to begin formatting x:
Press a key to start formatting your fixed disk. Press ^BREAK to exit. Remember that formatting eliminates all existing information on the disk. (FORMAT)

Print queue is empty
There are currently no files in the print queue. (PRINT)

Print queue is full
There are already 10 files in the print queue. (PRINT)

Printer lines per inch set
An instruction setting the printer vertical spacing to six or eight lines per inch was given. If this is unacceptable, an error message is displayed. (MODE)

Probable non-DOS disk
Continue (Y/N)?

The File Allocation Table id byte cannot be read. If you respond with a Y and did not specify the /F switch, CHKDSK displays possible remedies for the problem. (CHKDSK)

Processing cannot continue

This message preceded another message explaining why processing cannot continue. (CHKDSK)

Program size or number of segments
exceeds capacity of MS- LINK

The total size may not exceed 384 Kbytes, and the number of segments may not exceed 255. (MS-LINK)

Program too big to fit in memory

There is not enough memory available to allow the file containing the external command to be loaded. Specify a smaller value for BUFFERS=. (MS-DOS)

Requested partition access cylinders that are not available.

The location specified for an MS-DOS partition does not have sufficient space. (FDISK)

Requested size is greater than maximum available.

There is insufficient room on the fixed disk for an MS-DOS partition of the requested size. (FDISK)

Requested stack size exceeds 64K
Specify a size less than or equal to 64 Kbytes after the STACK SIZE: prompt. (MS-LINK)

Resident part of PRINT installed
Notification that PRINT is installed and ready to process print commands. (PRINT)

Resident portion of MODE installed
The part of MODE used for nonscreen-setting functions has been loaded into memory. (MODE)

Sector size too large in file <*filename*>
You specified a device driver with a sector size larger than the one that was previously defined. (MS-DOS)

Stack size exceeds 65535 bytes
Stack size must be less than or equal to 65535. (MS-LINK)

Syntax error
You made an error in entering a command. Check what you entered and retype it correctly. (MS-DOS)

Segment size exceeds 64K
64 Kbytes is the addressing system limit. (MS-LINK)

Symbol defined more than once
MS-LINK found two or more modules that define a single symbol name. (MS-LINK)

Symbol table capacity exceeded
Names typed were too long, or too many were specified, exceeding the limit of approximately 50 Kbytes. (MS-LINK)

Target diskette may be unusable
Unrecoverable read, write, or verify error messages cause this message to be displayed after DISKCOPY completes. The target diskette may have an incomplete copy because of the error. (DISKCOPY)

Target diskette write protected
Correct, then strike any key

You attempted to copy to a write-protected diskette. (DISKCOPY)

Terminate batch job (Y/N)?

This message is produced when ^BREAK is pressed during the processing of a batch file. Enter Y to stop the batch file processing. Entering N ends only the command that was executing when ^BREAK was pressed. The batch resumes processing with the next command in the file. (MS-DOS)

The current active partition is x

This is the partition that is now active. (FDISK)

The last file was not restored

Either you terminated RESTORE before it finished, or there was not enough disk space. The partially restored file has been erased. (RESTORE)

There was/were
number errors detected

This message is displayed for your information at the end of a link session. (MS-LINK)

Too many external symbols in one module

The limit is 256 external symbols per module. (MS-LINK)

Too many groups

The limit is 10 groups, including DGROUP. (MS-LINK)

Too many libraries specified

The limit is eight libraries. (MS-LINK)

Too many overlays

The limit is 64 overlays. (MS-LINK)

Too many public symbols

The limit is 1024 public symbols. (MS-LINK)

Too many segments or classes
The limit is 247 (segments and classes taken together). (MS-LINK)

Total disk space is xxxx cylinders.
This is a notification of how much disk space there is. (FDISK)

Track 0 bad-disk unusable
If track 0 is unusable, the required system information cannot be placed on the disk. (FORMAT)

Tree past this point not processed
CHKDSK aborted because track 0 is bad. (CHKDSK)

Unable to create directory
Either the root directory is full, a specified pathname was not found, or you specified a directory that already exists. (MKDIR)

Unable to Write BOOT
The BOOT record could not be written because track 0 is bad. (FORMAT)

Unexpected end-of-file on library
There is probably an error in the file. (MS-LINK)

Unexpected end-of-file on VM.TMP
The diskette containing the VM.TMP file was removed. (MS-LINK)

Unrecognized command in CONFIG.SYS
There is an incorrect command in CONFIG.SYS. Correct the command and try again. (MS-DOS)

Unrecoverable format error on target
Target diskette unusable
There is an unrecoverable error on the target diskette. There is no unusable data on the diskette. (DISKCOPY)

Unrecoverable read error on drive X
Track xx, side x

The data could not be read from the indicated track and side after four attempts. (DISKCOMP)

Unrecoverable read error on source
Track xx, side x

Data from the source diskette could not be read after four attempts. The copying process continues, but the copy may be incomplete. (DISKCOPY)

Unrecoverable verify error on target
Track xx, side x

Write operation to the target diskette could not be verified after four attempts. The copying process continues, but the copy may be incomplete. (DISKCOPY)

Unrecoverable write error on target
Track xx, size x

Data could not be written to the target diskette after four attempts. The copying process continues, but the copy may be incomplete. (DISKCOPY)

Unresolved externals: list

The external symbols listed have no defining module among the modules or library files specified. (MS-LINK)

VM. TMP READ ERROR

Disk read error; try again. (MS-LINK)

VM.TMP is an illegal filename and has been ignored

An object file cannot be named VM.TMP. This message is a warning only. (MS-LINK)

VM.TMP has been created
Do not change disk in drive (A: or B:)

MS-LINK has created the VM.TMP file because available memory has been exceeded. Do not remove the diskette from the drive until the link session ends. (MS-LINK)

Volume in drive X has no label
When you formatted a partition or diskette, you did not give a volume name. (MS-DOS)

Volume label (11 characters, ENTER for none)?
To include a volume label, enter it at this time. Otherwise, press ENTER. (FORMAT)

Warning! All data in the MS-DOS partition will be DESTROYED. Do you wish to continue? [N]
Proceed only if you want to delete the MS-DOS partition. (FDISK)

Warning-directory full
There is not enough room in the directory to recover any more files. Copy some files to another disk and try again. (RECOVER)

Warning! Diskette is out of sequence
Replace the diskette or continue
Strike any key when ready
You attempted to restore files with the diskettes out of order. (RESTORE)

Warning! File xx
is a read-only file
Replace the file (Y/N)?
This prompt is displayed if you used the /P switch. (RESTORE)

Warning! File xx
was changed after it was backed up
Replace the file (Y/N)?
The named file on the back-up diskette has an earlier date and time than the one on the fixed disk. Entering Y causes it to be replaced with the back-up copy. (RESTORE)

Warning! No files were found to back up
No files were found with the specified names. (BACKUP)

Warning! No files were found to restore

No files were found with the specified names. (RESTORE)

Warning: no stack segment

None of the specified object modules contains a statement allocating stack space, but you typed the /STACK switch with a positive numeric value. (MS-LINK)

WARNING - - Read error on EXE file

The input file contains an error that may cause the output file to be unusable. (EXE2BIN)

Warning: Segment of Absolute or Unknown Type

There is a bad object module, or an attempt has been made to link modules that MS-LINK cannot handle (eg, an absolute object module). (MS-LINK)

Warning! The data in drive C: will be destroyed.
Do you want to continue the format (Y/N)? [N]

Formatting the disk will erase all information currently stored on it. Respond Y to proceed, N to quit (N is the default). (FORMAT)

Warning? The data in the MS-DOS partition will be destroyed. Do you want to remove the MS-DOS partition (Y/N)? [N]

Proceed only if you want to delete the MS-DOS partition. (FDISK)

Write Error in TMP File

No more disk space remains to expand the VM.TMP file. (MS-LINK)

Write Error on RUN File

Usually, there is not enough disk space for the run file. (MS-LINK)

Write fault error writing device LPT1
Abort? Retry? Ignore?

MODE command (option 1) is unable to set the printer mode because of an I/O error, out of paper, POWER OFF, or time-out condition. An error code may have been returned that sends a control character to the printer; if so, the control character did not take effect. (MODE)

x is not a choice. Enter a choice.

You gave an invalid response to a prompt. (FDISK)

xxxx bytes freed

Disk space designated as allocated was not allocated. The space was freed and is available. (CHKDSK)

xxxx error on file yyyy

This is a printer message describing an error. The printing halts. (PRINT)

xxx lost clusters found in *yyy* chains

CHKDSK has located lost information and asks you how to proceed. (CHKDSK)

***** Backing up files to diskette xx *****

This message precedes a list of files being backed up. (BACKUP)

***** Files were backed up *mm/dd/yy* *****

The listed files were backed up on the date shown. (BACKUP)

*****Restoring files from diskette xx *****

This message precedes a list of files being restored. (RESTORE)

— MORE —

Press any key to proceed to the next data screen. (MORE)

10 Mismatches — ending compare

Files being compared contained 10 mismatches. No further comparisons are made. (COMP)

INDEX

SPECIAL CHARACTERS

* (asterisk; also wild card character) 1-17, 1-18, 4-52, 4-59
  current line (EDLIN). 5-20
  prompt (EDLIN) 5-1, 5-22, 5-57
  . (wild card file reference). 1-19, 4-49, 4-59, 4-61
@ (at sign). 5-57
\ (back slash). 1-15, 3-3, 5-55
  (blank space). 6-12, 7-5
{ } (braces) 1-13
[ ] (brackets) 1-13
¦ (broken bar) 1-13, 1-15, 4-15, 4-93
^(caret) 1-2, 5-7, 7-3
, (comma) 1-13, 5-29, 7-5
: (colon). 1-13, 4-19, 4-74, 4-109
$ (dollar sign). 4-93, 4-94
. . . (ellipses) 1-13, 4-19
= (equal sign) 1-13, 4-93, 7-19
> (greater-than sign) 1-15, 4-13, 4-93
- (hyphen) 4-48, 7-16, 7-32
< (less-than sign). 1-15, 4-14, 4-93
– (minus sign) 5-3
% (percent sign). 4-9, 4-107
. (period; also root directory notation). 3-4, 4-103, 4-109, 5-10, 5-29
.. (parent directory notation). 3-4, 4-103
+ (plus sign) 5-3, 6-12, 6-16
# (pound sign) 4-19, 5-10
? (question mark; also wild card
  character) 1-17, 1-18, 4-52, 5-40, 5-43
' (quotation mark - single) 7-8
" (quotation marks - double) 7-8
; (semi-colon) 4-19
/ (slash). 1-13, 4-5, 4-48, 6-13
^ALT + DEL (system reset). 1-3, 1-22, 5-8
^BREAK (abort). 1-3, 4-5, 4-16, 4-29, 4-30, 4-76,
  4-79, 4-85, 4-100, 5-8, 6-17, 7-3

^ENTER (continue line) . . . 1-3
^NUM LOCK (suspend operation) . . 1-3, 4-5, 4-85, 4-114, 5-8, 7-3
^PRTSC (echo to printer) . . . 1-2, 4-114, 5-8
^V (EDLIN) . . . 5-4
^Z (EOF marker) . . . 5-3, 5-4, 5-9, 5-22, 5-58
8086/8088 computers . . . 1-1
8086/8087/8088 mnemonics . . . 7-9
8087 opcodes . . . 7-11

A

A> prompt . . . 4-6, 5-1
Abort (see ^BREAK)
addresses, memory . . . 6-1, 6-3
ANSI.SYS device driver . . . 4-19, 4-50
APPEND command (EDLIN) . . . 5-2, 5-13, 5-48
append lines . . . 5-26
argument . . . 4-4
ASSEMBLE (A) command (DEBUG) . . . 7-4, 7-9
assembly language programs . . . 6-3, 6-4, 6-8
ASSIGN command . . . 4-26
asynchronous communications . . . 4-47, 4-84
  initializing . . . 4-82
AUTOEXEC.BAT file . . . 1-26, 4-7
automatic program execution . . . 1-26
automatic response file . . . 6-1
AUX: (device name) . . . 1-17, 4-47, 4-91

B

BACKSPACE key . . . 1-2, 5-6
BACKUP command . . . 4-27, 4-99
backup copies . . . 1-5, 4-27
.BAK file extension . . . 5-4, 5-22, 5-37
BASIC . . . 6-4, 6-8

.BAT file extension 4-7
batch file 4-7, 4-74
  execution 4-10, 4-88, 4-97
  parameters 4-9, 4-10, 4-105, 4-107
batch processing commands 4-7, 4-11
baud rate 4-83
.BIN file extension 4-62
BREAK command 4-16, 4-30
BUFFERS command 4-17, 4-31
buffers
  disk 4-16
  memory 4-31

C

CAPS LOCK key 1-2, 5-6
characters, valid file specification 1-15
CHDIR (CHANGE DIRECTORY) command 3-6, 3-8, 4-32
CHKDSK (CHECK DISK) command 4-33
class name, declared 6-7
classes, memory 6-3
CLS command 4-36
COBOL 6-4, 6-8
COM file extension 4-62
COM1: (device name) 1-17, 4-47, 4-91
COM2: (device name) 1-17, 4-47, 4-91
combine types (MS-LINK) 6-5
command
  characters (MS-LINK) 6-16
  default values 4-4
  files 1-11
  parameters 4-4
  piping 4-15
  processor 4-104, 4-106
  prompts (MS-DOS) 4-93
  prompts (MS-LINK) 6-11, 6-18, 6-19
  syntax 1-13

COMMAND.COM file 1-11, 1-20, 4-7, 4-16, 4-60, 4-106, 4-110

commands

  configuration 4-16

  directory 3-6

  external 2-14, 4-3

  internal 2-14, 4-2

  rules for 4-5

communications protocol parameters 4-83

common segment combine types 6-5, 6-6

COMP (COMPARE) command 4-37

COMPARE (C) command (DEBUG) 7-4, 7-12

compiling programs 6-1, 6-2, 6-3

CON: (device name) 1-17, 4-47, 4-91

concatenation 4-39, 4-43, 4-45

CONFIG.SYS file 4-16, 4-19, 4-31, 4-65, 4-106

configuration commands 4-16

CONTROL key (see CTRL key)

COPY command 2-14, 4-39, 4-56, 4-59

  EDLIN 5-14, 5-22

CPU register 7-1

CTRL (CONTROL) key 1-2, 5-7

CTTY command 4-47

cursor keys

  CURSOR DOWN 5-7

  CURSOR LEFT 5-7

  CURSOR RIGHT 5-7, 5-8, 5-50

  CURSOR UP 5-7

cursor positioning 4-19, 4-20

cylinders, fixed disk 2-5

D

d: (drive specifier) 1-13

DATE command 4-8, 4-48

date
  display 4-7, 4-8, 4-94
  display and change 4-48
  entering 1-23
  file 4-46
DEBUG
  commands summary 7-4
  control characters 7-3
  editing keys 7-5
  entering commands 7-3
  parameters 7-5
  pseudo-instructions 7-10
declared class name 6-7
DEL (DELETE) command 1-19, 4-49, 4-61, 4-103
DEL (DELETE) key
  EDLIN 5-9, 5-53
  MS-DOS 1-3, 1-4
DELETE (D) command (EDLIN) 5-17, 5-23
delete lines 5-17
DEVICE command 4-16, 4-50
device driver (see ANSI.SYS)
device names (see also reserved device names) 1-17
DGROUP (MS-LINK) 6-14
dictionary-indexed search 6-13
DIR (DIRECTORY) command 3-6, 3-7, 4-51, 4-103
directories
  changing 3-6, 3-8, 4-32
  creating 4-70, 4-77
  deleting 3-9, 4-103
  displaying 3-10, 4-51, 4-109
  parent 3-1, 3-4, 4-103
  root 3-1, 4-16, 4-103
  searching 4-86
  structure 3-1
  structuring (hierarchical) 3-2
  subdirectory 3-1, 3-4, 3-6, 4-77
  tree-structured 3-2
  working 3-1, 3-4

directory commands 3-6
disk buffers 4-31
disk drive
  designators 1-1, 1-14
  fixed 2-1, 2-3
  source 4-6
  target 4-6
DISKCOMP (DISKETTE COMPARE)
  command 1-8, 4-26, 4-37, 4-54
DISKCOPY (DISKETTE COPY) command . . . 1-5, 4-26, 4-57, 4-71
diskette 1-1
  copying 4-57
  double-sided 1-14
  fragmentation 4-58
  single-sided 4-71
disks 1-1
  fixed (see fixed disk)
  volume label 2-12, 4-70, 4-117
display
  external monitor 4-80
  file 5-29
  options 4-19, 4-22, 4-80
drive designator 1-1, 1-14, 4-5
/DSALLOCATE switch 6-14
dual-drive system 1-1
dummy module 6-8
DUMP (D) command (DEBUG) 7-4, 7-13
dynamic (re)numbering 5-2, 5-46

E

ECHO command 4-11, 4-60
EDIT command (EDLIN) 5-20
editing buffer 5-48
EDLIN
  commands, summary 5-5
  editing functions 4-114, 5-5

- interline editing 5-5, 5-9
- intraline editing 5-5, 5-6
- key functions 5-6, 5-49

END command (EDLIN) 5-2, 5-10, 5-22, 5-37
END key 5-7
ENTER (E) command (DEBUG) 7-4, 7-16
ENTER key 1-2, 1-4, 4-5, 5-6
EOF marker (see ^Z)
ERASE command 4-49, 4-61
ESCAPE (ESC) key
- EDLIN 5-9, 5-55
- MS-DOS 1-2, 1-4, 4-76

.EXE file extension 4-62, 6-1, 6-9, 6-12, 6-15, 7-24
EXE2BIN command 4-62
external
- commands 2-14, 4-2, 4-3
- files 4-2
- references 6-2

F

F1 key
- EDLIN 5-7, 5-8, 5-50, 5-51, 5-53
- MS-DOS 1-3

F2 key
- EDLIN 5-8, 5-51, 5-54
- MS-DOS 1-3

F3 key
- EDLIN 5-9, 5-52, 5-53, 5-54, 5-55
- MS-DOS 1-3

F4 key
- EDLIN 5-9, 5-54
- MS-DOS 1-3

F5 key
- EDLIN 5-9
- MS-DOS 1-4

F6 key (EDLIN) 5-58

F7 through F-10 keys (EDLIN) 5-9
FDISK utility 2-1, 2-2, 2-5, 2-9, 2-15, 2-17, 4-64
file control block (FCB) 7-28
file extensions 1-14, 1-15, 4-2, 4-5
  see also —
    .BAK
    .BAT
    .BIN
    .COM
    .EXE
    .LIB
    .MAP
    .OBJ
file specifications 1-14, 1-15, 4-4
  illegal 1-16
  MS-LINK 6-9
  valid 1-14, 1-15
  valid characters 1-15
filename 1-14
FILES command 4-17, 4-65
files
  creating 5-24
  deleting 4-49, 4-61
  displaying 4-114
  formatting 4-70
  hidden 4-71, 4-108
  library 6-1, 6-9, 6-12
  listing
    all 3-7, 4-52
    selected 4-53
  MS-DOS 1-11
  protecting 1-21
  response 6-1, 6-11, 6-18, 6-22
  system 1-11, 4-70, 4-108
  temporary 6-2, 6-10
  used by MS-LINK 6-9
filespec (file specification, in syntax) 1-14
FILL (F) command (DEBUG) 7-4, 7-18
filter commands 4-14, 4-64

filters 1-13
FIND command 4-14, 4-66
fixed disk
  backing up 4-27, 4-99
  capacity 1-1, 2-1
  drive designators 1-1
  partitions. 2-2, 2-3, 2-5, 4-64
  selecting 2-21
flags
  register display 7-31
  user 7-20
floppy diskette (see diskette)
FOR command 4-11, 4-68
FORMAT command 2-12, 4-70, 4-108
formatting
  diskettes 1-1, 4-70
  fixed disk (see FDISK and FORMAT)
  partitions 2-12
  single-sided diskettes 4-71
FORTRAN 6-4, 6-8, 6-14
function calls 4-15

G

GO (G) command (DEBUG) 7-4, 7-6, 7-19, 7-39
GOTO command 4-11, 4-74
global symbols 6-15
GRAPHICS command 4-75
graphics display (see ANSI.SYS)
groups, memory 6-3, 6-4

H

HEX (H) command (DEBUG) 7-4, 7-21
HEX file 7-24
hidden files 4-71, 4-108

hierarchical directories . . . 3-2
/HIGH switch . . . 6-14
high-resolution graphics . . . 4-75

I

IF command . . . 4-11, 4-76
INPUT (I) command (DEBUG) . . . 7-4, 7-22
input
  device . . . 4-47
  files (MS-LINK) . . . 6-9
  redirection, from keyboard . . . 4-13, 4-14
INS (INSERT) key
  EDLIN . . . 5-56
  MS-DOS . . . 1-4
INSERT command (EDLIN) . . . 5-22, 5-24
insert mode . . . 5-56, 5-58
instruction pointer . . . 7-2, 7-20
internal commands . . . 2-14
interline editing commands (EDLIN) . . . 5-5, 5-9
  parameters . . . 5-10
intraline editing commands (EDLIN) . . . 5-5, 5-6
italics, in syntax . . . 1-13

K

key functions (EDLIN) . . . 5-6, 5-7, 5-8, 5-9
key reassignments . . . 4-19
keyboard (MS-DOS) . . . 1-2, 1-3, 1-4

L

.LIB file extension . . . 6-9, 6-12
libraries . . . 6-11
  prompt . . . 6-11, 6-12

library files 6-9
library search 6-1
/LINENUMBERS switch 6-15
linker
  utility 6-1
  virtual 6-1
LIST command (EDLIN) 5-1, 5-29
list file 6-9
  prompt 6-12
LOAD (L) command (DEBUG) 7-4, 7-6, 7-23, 7-26, 7-27
loading, MS-DOS 1-22
LPT: (device name) 1-17
LPT2: (device name) 1-17
LPT3: (device name) 1-17

M

machine language 6-1
.MAP file extension 6-9, 6-12
/MAP switch 6-15
MKDIR (MAKE DIRECTORY) command 3-6, 4-77
mnemonics 7-9
MODE command 4-78
MORE command 4-14, 4-85
MOVE command
  DEBUG (M) 7-4, 7-25
  EDLIN 5-32
MS-DOS
  commands
    external 2-14, 4-2, 4-3
    internal 2-14, 4-2, 4-11
  diskette 1-5
    comparing 1-8
      dual-drive procedure 1-13
      single-drive procedure 1-8
    copying 1-5
      dual-drive procedure 1-7
      single-drive procedure 1-5

    files 1-8
    loading 1-22
MS-LINK
    command characters 6-16
    files 6-9
    invoking 6-18
    switches 6-11, 6-13

N

NAME (N) command (DEBUG) 7-23, 7-24, 7-26, 7-39
notation, syntax 1-13
NUM LOCK (NUMBER LOCK) key 5-7

O

operating system
    changing 2-15
    installing 2-2, 2-5, 2-9
    transferring files 4-108
.OBJ file extension 6-9
object
    code 6-1
    file 6-9
    modules 6-1, 6-11
    prompt 6-11
opcode synonyms 7-11
OR statement, in syntax 1-13
OUTPUT (O) command (DEBUG) 7-4, 7-29
output
    device 4-47
    redirection 4-13, 4-78
overlays 6-1

P

PAGE command (EDLIN) 5-34
parameters
  communications protocol 4-83
  DEBUG 7-5
  MS-DOS 4-4
  MS-LINK 6-13
  replaceable batch 4-9, 4-105, 4-107
parent directory notation (see also directories, parent) 3-4
partitions
  fixed disk 4-64
  formatting 2-12
  installing 2-2, 2-3, 2-5, 2-9, 4-64
  listing data 2-20
  modifying active 2-15
  removing 2-17
  volume label 2-12
Pascal 6-4, 6-8, 6-14
PATH command 4-86
pathnames 3-3
PAUSE command 4-7, 4-12, 4-88
/PAUSE switch 6-15
pipes 1-13
piping, command 4-15
PGDN (PAGE DOWN) key 5-7
PGUP (PAGE UP) key 5-7
PRINT command 4-90
print, graphics display 4-75
print mode 4-78
private segment combine types 6-5
PRN: (device name) 1-17
PROMPT command 4-93
PRTSC (PRINTSCREEN) key 1-2, 5-8
pseudo-instructions (DEBUG) 7-10

public
  segment combine types 6-5
  symbols 6-15

Q

QUIT command
  DEBUG (Q) 7-4, 7-5, 7-30
  EDLIN 5-2, 5-10, 5-17, 5-23, 5-37

R

RECOVER command 4-95
REGISTER (R) command (DEBUG) 7-4, 7-19, 7-31
register indirect commands 7-11
REM (REMARK) command 4-7, 4-12, 4-97
REN (RENAME) command 4-98
REPLACE command (EDLIN) 5-3, 5-38
replaceable parameters, batch file 4-9, 4-105, 4-107
reserved device names 1-17
  see also —
    AUX:
    COM1:
    COM2:
    CON:
    LPT:
    LPT2:
    LPT3:
    PRN:
response file (MS-LINK) 6-1, 6-11, 6-18, 6-22
RESTORE command 4-99
RMDIR (REMOVE DIRECTORY) command 3-6, 3-9, 4-103
root directory (also see directories, root) 3-1
run file 6-1, 6-9, 6-14, 6-15
  prompt 6-11, 6-12

S

screen display options (see display options)
scrolling 5-34
SEARCH command
  DEBUG (S) 7-4, 7-34
  EDLIN 5-3, 5-43
segment name 6-3
segment registers 7-2, 7-20
serial interface, printer 4-84
SET command 4-104
SHELL command 4-17, 4-106
SHIFT command 4-9, 4-12, 4-107
SHIFT keys 1-2, 5-6, 5-7
single-drive system 1-5, 1-8, 4-34
single-sided diskettes 4-70
SORT command 4-14, 4-108
source code 6-1
SPACEBAR 5-6, 5-51, 7-16
stack pointer 6-5, 7-3, 7-20
stack segment combine type 6-5
/STACK switch 6-16
subdirectory 3-1
  creating 3-6, 4-77
  notation 3-4
switches
  MS-DOS 4-4
  MS-LINK 6-11, 6-13
syntax notation 1-13
SYS command 4-110
system reset (see also ^ALT + DEL) 1-22

T

TAB key 5-7
temporary file (see also VM.TMP) 6-2

TIME command 4-8, 4-111
time
  displaying 4-94
  entering 1-23, 4-111
TRACE (T) command (DEBUG) 7-5, 7-6, 7-35, 7-39
TRANSFER command (EDLIN) 5-46
TREE command 3-6, 3-9, 4-113
tree-structured directories 3-2
TYPE command 4-114

U

UNASSEMBLE (U) command (DEBUG) 7-5, 7-6, 7-36
upper-case words, in syntax 1-13

V

variables, in syntax (see italics)
VER (VERSION) command 4-115
VERIFY command 4-116
virtual
  link 6-1
  memory file 6-1
VM.TMP file 6-2, 6-10, 6-16
VOL (VOLUME) command 4-117
volume label
  assigning 2-12, 4-70
  displaying 4-117

W

wild card
  characters. . . . 1-17, 4-40, 4-44, 4-49, 4-52, 4-61, 4-95, 4-98, 4-100
  see also —
    * (asterisk)
    ? (question mark)
working directories 3-1
  changing 3-8
  displaying. 3-6, 3-8
WRITE command
  DEBUG (W) 7-5, 7-6, 7-26, 7-27, 7-39
  EDLIN 5-13, 5-48
write-protected diskettes 1-5

USER SURVEY*

COMPAQ® Computer Corporation is dedicated to providing you with quality documentation. However, this can only be accomplished with your help. We are, therefore, requesting that you take a few minutes to read through this survey form. Then, when you have used the manual, please answer the questions listed below – and drop this prepaid form in the mail.

Thank you for taking the time to assist us.

	Excellent	Good	Fair	Poor
1. Organization of guide.	________	______	______	______
2. Usefulness of examples.	________	______	______	______
3. Clarity of information.	________	______	______	______
4. Completeness of information.	________	______	______	______
5. Layout of manual (ease of use).	________	______	______	______
6. Appropriateness to your experience level.	________	______	______	______
7. Overall rating.	________	______	______	______

8. How much experience have you had with personal computers?

none ____________
0-6 months ____________
6 months - 1 year ____________
1 year or longer ____________

9. Please list any errors found in this guide:

Error	Page
______________________________	______________
______________________________	______________
______________________________	______________
______________________________	______________

10. Please provide any additional comments or suggestions below:

____
____
____

(Optional)
Name______________________________________
Company___________________________________
Street____________________________________
City______________________State__________Zip__________

*MS™-DOS Version 2 Reference Guide

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 400 HOUSTON, TEXAS USA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPAQ Computer Corporation
Attention: Customer Relations
20555 FM 149
Houston, Texas 77070

Fold here

Tape

Please do not staple

Tape